Component-based Development of MILLS: A Case Study in the development of an Inter-library Loan Software System

Naresh Kumar Agarwal, Danny C. C. Poo and Teo Keng Yong
School of Computing, National University of Singapore
3 Science Drive 2, Singapore 117543
{naresh, dpoo, teokengy}@comp.nus.edu.sg

Abstract

Component-based Software Development (CBSD) is based on the idea of developing software systems by selecting appropriate off-the-shelf components and integrating them under a specified architecture. Such an approach prevents a software developer from having to reinvent the wheel, reduces software development cost and time, and promotes flexibility and maintainability. In this paper, we describe the application of CBSD in developing a system for automated inter-library loans between the libraries of two major universities of Singapore. Currently, inter-library loan (ILL) is a cumbersome and manual process. Our system helps promote resource-sharing and interoperability and reduces the difficulties involved in the current manual loan system between libraries. Specifically, we describe the CBSD approach, present the design and implementation of the MILLS (Managing Inter-library Loan Services) system, and describe how MILLS was developed taking the CBSD approach into consideration. The lessons learnt from this case should be of value to software developers.

1. Introduction

As computer-based application systems are being ingrained in business enterprises, including libraries and institutions, there is increasing pressure on those involved in software development for the systems to be developed quickly with higher flexibility and maintainability. Component-based Software Development (CBSD) is based on the idea of developing software systems by selecting appropriate off-the-shelf components and integrating them under a specified architecture. A component is defined as a piece of executable software with a published interface [1]. Components can either be developed by in-house developers or purchased readily from the market at a certain cost. CBSD prevents a software developer from having to *reinvent the wheel* all the time and helps reduce software development cost and time, and promotes flexibility and maintainability. It helps channel the resources and expertise of a large pool of software developers by providing reusable components, that will benefit the software engineering community at large, and its large number of stakeholders. Thus, it can potentially be used to reduce software development costs, assemble systems rapidly, and reduce the spiraling maintenance burden associated with the support and upgrade of large software systems [2].

There are four primary stakeholders of Component-based Software Development:

1. Framework developers,
2. Component developers,
3. Application assemblers and
4. Customers

Framework developers create the infrastructure for components to interact [3]. Sun Microsystems, through the Sun Java System (J2EE) Portal Server [4] provides a Java (J2EE) enterprise environment that allows users to create and manage portal channels. In addition, the Java platform offers an efficient solution to portability and security problems through the use of portable Java byte codes [5]. *Component developers* identify suitable domains and develop new components for systems [3]. Such developers could come from within freelance developers, information systems (IS) departments of institutions or an external organization specializing in component development [2]. *Application assemblers* select the specific components suitable to the system under development

and assemble them into the integrated system [2][3]. *Customers* are users that employ the component-based system to perform the task [2].

In this paper, we describe the application of Component-based Software Development in developing a system for automated inter-library loans between the libraries of two major universities of Singapore.

In Section 2, we will look at the resource-sharing environment and the current manual inter-library loan (ILL) process between libraries. The challenges and the opportunities leading to the motivation for MILLS are also discussed. Section 3 discusses MILLS, the requirements gathering and the system architecture. In Section 4, we see how MILLS uses a Component-based Software Development approach in its implementation, and the components it uses. In Section 5, we highlight the key lessons learnt from this case study. We conclude the paper in Section 6.

Let us now look at inter-library loan.

2. Resource-sharing Environment and Inter-library Loans

Inter-library Loan (ILL) service in an academic library may operate under different names, such as inter-library borrowing, inter-library lending, inter-library services, document delivery, resource-sharing, etc. This paper will use inter-library loan or ILL to refer to these services. In most academic libraries, an eligible ILL patron is anyone with a library card in need of a book that the library doesn't have or is loaned out. In most universities, the faculty members and graduate students are the heaviest users of ILL services [6].

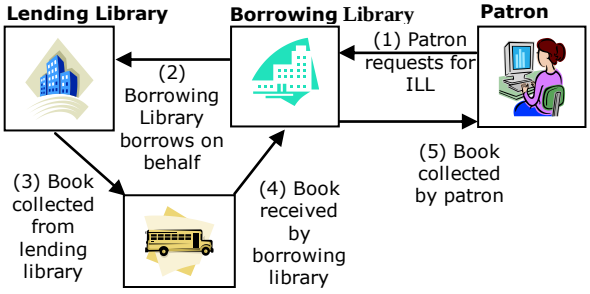

Figure 1. Inter-library Loan (ILL) Process

Figure 1 shows the ILL process. If a book or item needed by a library patron is not available or is checked out, a registered borrower may have the option of requesting the item from another library (lending library). An inter-library loan works like an ordinary library loan except that now the borrowing library becomes the 'patron' of the lending library.

Resource sharing generally refers to co-operative activities between libraries designed to maximize service while minimizing costs [7]. Libraries around the world are working together to form a resource-sharing environment. Under such an environment, librarians are looking into an automation system that could assist in their work. Patrons, on the other hand, are looking for better services. Google[1] and Amazon[2] provide their users with simplicity of information discovery and a one-stop shopping experience. Library patrons are expecting library services to be 'Amazoogle-like' i.e. providing services on *their* terms instead of the terms of the library. But these days, with rising costs and shrinking library budgets, libraries are unable to keep up with such expanding patron needs or exploding publishing output [8]. Even if all libraries were able to spend as much as they wanted on library resources, there would always be some resources, which they need to obtain on behalf of the patrons on an ad-hoc basis. With the growth of Email and the Internet, libraries are able to provide services to one another and share resources. Many are also trying to form a community collection in order to assist each other in the reduction of duplicated resources and to maximize the use of the limited amount of funds each library is allocated [9]. It also becomes clear that libraries have to cooperate together regionally, nationally and internationally in order to achieve any real progress [10]. Additionally, the idea of having a union catalogue (which lists all the books in participating libraries) could also promote resource sharing among libraries [10], as patrons are able to have access to different catalogues. Therefore, inter-library loan (ILL) plays an important role in the creation of a resource-sharing environment [11].

In Singapore, inter-library loan has been a mutual agreement service between many tertiary institutions. Certain standards and procedures have been defined for processing an ILL request. Internet has also changed the cooperation involved in an ILL request from national to international [11]. However international inter-library loans are posed with technical and logistical problems for union catalogues. The ILL process between libraries in Singapore is largely manual and cumbersome. A primary reason for lack of automation is because major libraries have

[1] www.google.com
[2] www.amazon.com

their own IT systems developed, which do not talk to the systems implemented in other libraries. The difference may be in the operating system, programming language and message standards. Over the years, many systems have been developed to assist in the resource-sharing environment. However such systems usually lack in uniform standards, communication with other systems or interoperability. Foo and Lim [12] describe the implementation of an integrated ILL system in the Nanyang Technological University (NTU), Singapore. The system was web-based and was developed to fulfill NTU's ILL requirements. As it didn't involve other libraries, it didn't take off and the librarians later abandoned it. Thus, from a system viewpoint, it is necessary to integrate ILL systems involving different libraries, local library systems and to determine the end patron's ILL system and rights [11]. There is a need to have a common communication channel to support the increasing demand of resource-sharing between libraries. Technology can be a great facilitator in this regard. With web services, library data can be more easily exposed for consumption and reprocessed for other applications systems, reaching out to wider audiences and serving a greater variety of uses. Component-based Software Development (CBSD), as discussed, can help improve flexibility and maintainability and also cut time and resources for libraries systems. A standardized protocol for information exchange (e.g. Z39.50) is also mandatory to facilitate access of libraries catalogues over the Internet. Such a protocol will help different library systems to talk to each other. Therefore, a system is needed that not only helps to manage the ILL process but one, that is also able to provide interoperability between systems located within national or international boundaries. These expected benefits form the primary motivation to provide a new automated system to manage the ILL process under a resource-sharing environment.

3. Case Study: The MILLS (Managing Inter-library Loan Services) Project

Managing Inter-library Loan Services (MILLS) is a project that aims to facilitate resource-sharing and automate the inter-library loan (ILL) process between libraries in Singapore. From different perspectives like automation, interoperability, distributed computing and services, MILLS provides several advantages over the manual process including reduction of human errors, eliminating paper filling or processing and improving the user accessibility to different sources. The system architecture and implementation of MILLS closely follows the Component-based Software Development (CBSD) approach, which can potentially improve the flexibility and maintainability of MILLS.

The requirements were gathered from two Singapore libraries – the Central Library of the National University of Singapore (NUS) and the Lee Wee Nam Library (Library A) of the Nanyang Technological University (NTU). The design of MILLS is based on the system requirements gathered from the two libraries, and aims to help improve the ILL processes of both the libraries.

3.1. Requirements Gathering

The libraries of NTU and NUS were approached to gather the requirements for MILLS. Both NTU Circulation Services Division and NUS Circulation Department have helped in explaining and walking through the ILL process. ILL systems in both libraries are still done manually and the librarians keep hard copies of any inter-library loan request. In NUS, ILL requests may be made on inter-library loan forms and submitted to the Circulation Department in Central Library or the specified Library concerned. Academic staff and graduate students are authorized to use the IIL services. The typical steps involved are:

1. A patron from a certain university (borrowing library) needs a book that is available in the library of another university (lending library).
2. Patron downloads ILL form, fills it and submits to librarian of the borrowing library (NUS allows submitting an e-form with the ILL request).
3. The librarian, on getting the ILL request, verifies patron particulars and his justification for the loan request.
4. Librarian checks on the availability and loan status of the book through a search into the OPAC (Online Public Access Catalogue) of the lending library.
5. Librarian then informs the lending library to hold the requested book.
6. All books requested through ILL are collected from the lending library twice a week.
7. On notification, patron comes to the library, collects the book and signs the ILL form.
8. Patron is sent a reminder on the due date.

9. Upon return, librarian seeks the patron feedback on the book, and recommendation for purchase(s).
10. Lending library must also keep track of all loan requests.

This manual process creates numerous disadvantages and brings inconvenience to both the librarians and the patrons. Though there are some enhancements in ILL management system implemented in other libraries or institutions to assist the librarian to automate the ILL process, however, neither NTU nor NUS has implemented such a system. One reason could be that these systems are difficult to implement under NUS' or NTU's context due to the functionality of any system being limited by the infrastructure that exists in the different locations [13]. But above all, many such systems are mostly stand-alone and there is no interconnectivity between systems. In other words, there are no means for these systems to exchange information and data. This is due to the interoperability problem of lacking a common standard [12]. Thus, loans to other libraries have to be handled manually by the local circulation module with the librarian acting as a dummy borrower [13].

Table 1 summarizes the problems faced by librarians in the current manual ILL system (left column). The column on the right lists the functionality required in MILLS to alleviate the problem.

Table 1. Problems in Manual System and Functionality needed

Problems and issues with current ILL process	Functionality required in MILLS
Patron fails to fill all necessary information	**E-form** ▪ Online form that allows patron to fill and submit the request. ▪ The form should be validated upon request submission.
Patron verification	**Login** ▪ A login mechanism that could verify and authenticate the patron, or ▪ Access to the library patron database.
Justification of the ILL request	**Patron profile** ▪ Based on the patron profile or past request, determine whether the patron can request for the particular book(s)
Manually checking availability of book from all different lending libraries a time-consuming process	**Search multiple libraries** ▪ Patron can search into multiple lending libraries and determine where the book(s) is located. ▪ Based on the request information (e.g. book title), search into different (selective) sources to determine the availability of the book(s) at the lending libraries. ▪ Present the librarian information on libraries that hold the particular book(s)
Keeping track of passive ILL request manually	**ILL-request-records management** ▪ Store every request in the database. ▪ Store, view and update request records. ▪ Sort, search and display the required records or information request on demand from the librarian. ▪ Ability to keep track of soon-to-be-due book(s)
Keeping track of ILL request from external library manually	**Communication between systems** ▪ A communication method that allows exchange of request information, status, etc. ▪ A common method like web services to provide this service. ▪ External request can be requested by invoking the services provided to submit the request.
Time spent sending individual emails on collection of books or reminders	**Automated reminder messages** ▪ Upon an event occurring, an email will be generated and forwarded to the patron automatically. ▪ An event could be triggered by the librarian (arrival of book(s)) or automated (due date).
ILL request reports	**Report generation** ▪ Based on requests stored in the database, present a break down of the ILL request. ▪ Present in many different forms that suits the management e.g. monthly, by subjects, most popular etc.
Acquiring the book	**Recommendations** ▪ Providing past ILL requests that can aid in making decision on acquiring a new book.
Different lending processes/ systems in NTU and NUS	**Administrative function** ▪ Provide the management of lending libraries (i.e. update, add, remove) that support z39.50. ▪ Select the way the decision of the request is to be made (e.g. request having to route through the librarian versus request being submitted directly to lending library upon patron submission).

3.2. System Architecture

The MILLS system adopts a distributed computing approach. Figure 2 shows two MILLS systems communicating with each other to exchange information between two libraries. The communication between these two systems is done using web services. This approach promotes exchange of information and services between these two libraries. As web services promote interoperability, other libraries could equip their own systems with web services to communicate with MILLS as long as the methods are defined. The patron or librarian can be using a web browser (e.g. Internet Explorer or Firefox) to access MILLS, which resides on the web server. MILLS is developed using J2EE and integrated in Sun Portal Server 6.0. It makes use of an embedded database residing in Sun Portal Server for storing all the necessary information. MILLS will be accessing an Email Server to perform email function and a Z39.50 client to perform Z39.50 query search.

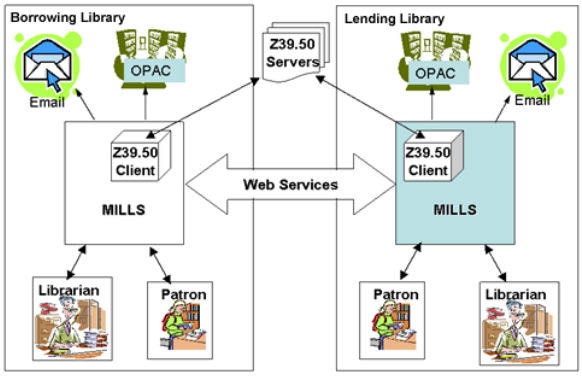

Figure 2. MILLS installed at both borrowing and lending library

4. CBSD Approach in MILLS

Figure 3 shows how MILLS follows a Component-based Software Development (CBSD) approach. The architecture shows three types of components – Java Application, J2EE Application and Portal Channels.

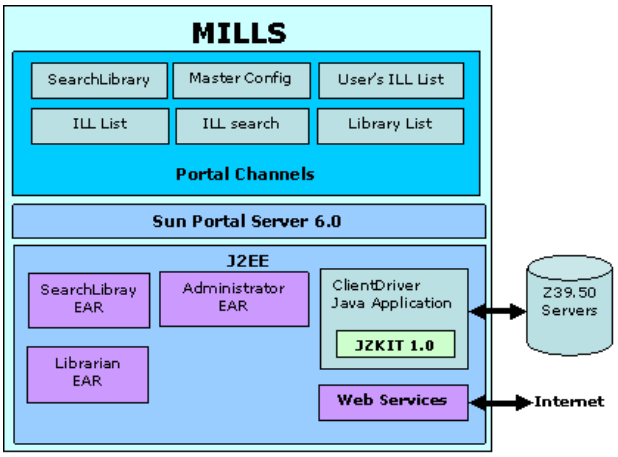

Figure 3. MILLS System Architecture – CBSD Approach

Table 2 summarizes the different components in the MILLS system.

Table 2. List of Components

Portal Channel components developed are:

- SearchLibrary
- Master Config
- User's Inter-library Loan List
- Inter-library Loan List
- Inter-library Loan List Search

Each portal channel component:

- Provides access to the different functions provided by the J2EE application
- Allows the user (i.e. patron, librarian or administrator) to select the functions that they are allowed to access and the interface whereby they access MILLS. Access control is enforced on the portal channels in order to restrict which users of MILLS can access which channels.

J2EE Application (EAR) components developed are:

- SearchLibrary
  - Provides the function to allow patron to perform search into Z39.50 servers.
  - Search supports single source request and even multiple sources request.
  - Also allows patrons to browse through the result list and select the book to request for an ILL and subsequently, monitor his ILL request list to keep him updated on the status of the request.
- LibrarianEAR
  - Provides functionality to monitor the ILL request from the patrons or external library.
  - Librarian can update that ILL request and message will be automatically sent to the requirement personnel.
- AdminEAR
  - provides functionality to perform administrative tasks like updating Z39.50 server information, email server configuration, web services communication configuration, etc.
- Web services

Java Application components developed are:

- Jzkit1.0
  - Java application that can perform Z39.50 search and retrieval.
- ClientDriver
  - Java application built on top of JZKIT to provide advanced functionality.

4.1. Implementation and Deployment

The MILLS system has been developed and presented to NUS and NTU libraries for further fine-tuning of the system. MILLS will be installed initially in both NUS and NTU libraries for a pilot run and for further evaluation. The system will be tested (e.g. black-box testing) and feedback gathered for fine-tuning. By gathering these feedbacks on the pilot system, additional requirements can be implemented to enhance the capability of MILLS.

MILLS deploys the CBSD system architecture and design (described in Section 3.2 and 4) and the requirements gathered (described in Section 3.1). In order to implement a CBSD approach for MILLS, a flexible software architecture that can outlive individual components [14] is required. Steps in engaging component-based software are [2][14]:

1. Qualification,
2. Adaptation,
3. Assembly and
4. Commercialization.

Qualification is the selection of components that are suitable for use in MILLS. E.g. JZKIT is an open-source Java Application that allows connection to the Z39.50 catalogue server to perform Z39.50 queries. As there were difficulties in finding certain components, these had to be developed from scratch. J2EE portal server allows the creation of such components or applications known as J2EE enterprise applications. *Adaptation* is the modification or alteration of the components that are selected during the qualification step. This is to ensure that components will synchronize with each other and meet the requirements of the MILLS system. E.g. JZKIT itself is not developed to support multiple connections at a generic level. A ClientDriver Java application is developed on top of JZKIT to produce the suitable component. *Assembly* is the integration of the components together to form the desired MILLS system. J2EE Portal Server provides the environment to create Portals Channels for access or linking the components selected or developed at the previous steps. *Commercialization* or updating of the components ensures that the components meet actual requirements and are reliable. Under J2EE Portal Server environment, each J2EE application or Java application can be updated and redeployed without affecting the whole MILLS or other components.

In the Introduction to this paper, we identified the four main stakeholders in CSBD as framework developers, component developers, application assemblers and customers. In the case of MILLS, framework developers will be the developers employed by Sun Microsystems who developed the Java Sun Portal Server and J2EE framework. This is because MILLS is developed using these. Component developers for MILLS will be the actual programmers or developers of MILLS. The application assemblers for MILLS could either be the developers or the technical support staff of the libraries that will implement MILLS. Customers will be the patrons, librarians or even the administrators of the library systems.

Let us now look at a few key lessons learnt from this case study.

5. Lessons Learnt

As this case involves multiple stakeholders and automates an existing work process, the lessons learnt should be most applicable to such situations.

5.1. Provide intermediate prototypes during Requirements Gathering

Requirements Gathering is a complex process where there is a constant need to reconcile the differences of the parties involved. A lot of times, the client does not know exactly what he wants. What is required only comes into light on seeing what new options are there to choose from. The new possibilities give rise to new requirements. Thus, presenting the client with enhancements at regular intervals will help refine the requirements so that the system increasingly gets closer to the desired functionality. Thus, this case highlights that requirements gathering is an ongoing process.

During the development phase of MILLS, a few rounds of presentation were conducted with the NUS and NTU librarians to demonstrate the development status of MILLS. The purpose of this presentation was to fine-tune the requirements and to gain feedback from the librarians for any additional information. A prototype of MILLS was presented whenever there were significant changes to the system. Based on each version of the system, the librarians were better able to articulate their requirements and decide on what they wanted. They were able to identify more issues, and offer further suggestions. The visual impact provided by the prototype played a key role in bringing these views and suggestions to surface.

5.2. Involve all stakeholders

It is important to understand and involve all the stakeholders in a project to ensure its success and subsequent adoption. In the case of ILL, involving the perspectives of multiple libraries is extremely important. For such a system to work, you must have the cooperation and sharing of information from multiple libraries. The system developed must be able to work seamlessly across the different systems of libraries. That is where interoperability is of prime importance.

A previous integrated ILL system was developed at NTU [12]. The system was ultimately abandoned because it was based on the concerns of a single stakeholder i.e. NTU. Had the concerns of other libraries in Singapore been addressed, there was greater likelihood of its success. MILLS has endeavored to fill this gap. The system developed is interoperable not just between the two libraries of NUS and NTU, the use of a common protocol (Z39.50), web services and reusable components has made the system scalable across libraries apart from these. Addition of a new system to MILLS is designed to be easy.

5.3. Put up a platform for stakeholders to reach common understanding

Where multiple stakeholders are involved, clear communication channels should be established and efforts should be made to provide a platform to reach a common understanding.

During the development of MILLS, the libraries of both NTU and NUS were involved at regular intervals, bilaterally as well as trilaterally. Such common meetings were useful not just in understanding existing processes but also helped in looking at new ways of doing things.

5.4. Utilize the components-based software development approach

By building the complete system from a pool of existing or ready-made components, the development time of MILLS was significantly reduced. There was no need to re-create those components that were already created by others or were available off-the-shelf. Each component created by the developer or purchased from the market can be used in other systems. E.g. a component that can search into multiple catalogues could be recycled in other systems like the Library main portal, student portal, etc. Each component was stress-tested in many applications to dig bugs or issues in order to improve that component. Changes made to one component will not affect the other components in the system. Additionally, replacement of obsolete components can be easily done. CBSD can provide an overall understanding of building systems in general and computer-based systems (whose significant part is software) in particular.

6. Conclusion

We have described how the Components-based Software Development (CBSD) approach was utilized for the development of an Inter-library Loan (ILL) system, MILLS. MILLS, Managing Inter-library Loan Services, was proposed and developed under the requirements gathered from NUS and NTU. The system has demonstrated the advantages of such

systems which not only improve the ILL work process but also fulfill the requirements specified in Table 1.

Automating any part of the ILL operations is influential in increasing efficiency and effectiveness, from having the patron creating the request to delivering electronic copies to their desktop. In 2000, the ILL Benchmarking Study [15] found that the turnaround time was 11.5 days, a significant improvement from 19 days [16]. Automation has enabled the ILL unit to accurately estimate dates of arrival for orders and to reduce delivery time [17]. Automated systems have also increased the fill rate as validation on the information filled by the patrons can be checked automatically. Additionally, due to prior verification of requests before submission to the lending library, the lending library receives lesser incorrect items. With the use of e-mail in automation ILL system, patrons can be readily informed on the arrival or delay of the loan at a faster service [17]. Having an automated system implies that the statistics of the requests can be captured and generated into useful reporting information. Subsequently, staff time is freed due to the drop in such time-consuming functions.

Application of CBSD to ILL will help ensure flexibility and maintainability in the system. Due to increased speed and efficiency of technology, the range and number of items loaded and received will soon increase over time [10]. The concept of using standards, technologies and automation is simple to explain but the difficulties reside in the implementation and encapsulation issues [18]. These factors will also be critical to allow communication across the different computer systems. MILLS has brought in the concept of automation and web services to improve the ILL services and allow better communication between different systems. From the patron point of view, having additional services (e.g. union catalogues) can help to ease their process of information retrieval. Examples are reducing the repetition in an initial short list display and providing guidance to subsequent exploration [11]. As for the librarian, certain tasks can be automated, thus providing a more efficient work process. Information can also be passed over to other libraries to facilitate communication.

Flexibility and maintainability have been primary motivators behind the development of MILLS. This was facilitated by Component-based Software Development, which is based on the assumption that certain portions of large software systems can be easily reused [2]. This approach guided the development of MILLS throughout. The lessons learnt from MILLS should be useful for other software systems using CBSD.

Acknowledgement

The authors are grateful to the Circulation Department of the NUS Central Library and the Circulation Services Division of the NTU Library for their valuable assistance in providing information on their inter-library loan process, and helping gather the requirements for MILLS.

References

[1] P. Vitharana, "Risk and Challenges of Component-based Software Development", *Communications of the ACM*, Vol. 46 Issue 8, Aug 2003, pp. 67-72.

[2] G. Haines, D. Carney, and J. Foreman, "Component-based Software Development / COTS Integration", *Software Engineering Institute*, Carnegie Mellon University, 1997, Retrieved 21 Jun 2006 from http://www.sei.cmu.edu/str/descriptions/cbsd_body.html

[3] A.I. Morch, G. Stevens, M. Won, M. Klann, Y. Dittrich, and V. Wulf, "Component-based technologies for end-user development", *Communications of the ACM*, Vol. 47 Issue 9, End-user development: tools that empower users to create their own software solutions, Special Issue: End-user development, Sep 2004, pp. 59-62.

[4] Sun website, "Sun Java System Portal Server", *Portal Services*, *Sun Microsystems*, 2006, Retrieved 22 Jun 2006 from http://www.sun.com/software/products/portal_srvr/

[5] X. Cai, M. R. Lyu, K.-F. Wong, and R. Ko "Component-based software engineering: technologies, development frameworks, and quality assurance schemes", Proceedings of *the Seventh Asia-Pacific Software Engineering Conference (APSEC'00)*, 5-8 Dec, 2000, Singapore, pp. 372-379.

[6] L. A. Hilyer, "Inter-library Loan Service from a Patron Perspective" in L.A. Hilyer, *Inter-library Loan and Document Delivery in the Larger Academic Library*, Haworth Press: Binghamton, NY, 2002, pp.19-26.

[7] N. Krym, and M. VanBuskirk, "Resource-sharing roles and responsibilities for CISTI: change is the constant", *Interlending & Document Supply*, Vol. 29 Issue 1, 2001, pp. 11-16.

[8] M. E. Jackson, "Re-examining the future of resource sharing", *Interlending & Document Supply*, Vol. 33 Issue 4, 2005, pp. 212-213.

[9] C. Boukacem, "Inter-library loan services and access to electronic resources in French university libraries: a marriage of reason", *Interlending & Document Supply*, Vol. 31 Issue 4, Nov 2003, pp. 218-227.

[10] J. Rachinger, "New developments in interlending in the Austrian National Library", National Library Visions, *Interlending & Document Supply*, Vol. 31 Issue 2, Jun 2003.

[11] J. Gatenby, "Inter-library loans and document delivery via EUCAT: a new PICA/OCLC initiative", *Interlending & Document Supply*, Vol. 31 Issue 2, Jun 2003, pp. 123-129.

[12] S. Foo and E.-P. Lim, "An Integrated web-based ILL system for Singapore libraries", *OCLC Systems & Services*, Vol. 14 Issue 1, 1999, pp. 24-34.

[13] J. Leeves, "Automation of ILL Management System", *Interlending & Document Supply*, Vol. 21 Issue 2, 1993, pp. 12-17.

[14] B. Berenbach, P. Spool, and D. Bitterle, "The application of modern software engineering practices to control engineering", *Journal of Advanced Manufacturing Systems*, Vol. 2 Issue 1, 2003, pp. 127-141.

[15] National Resource Sharing Working Group, "Inter-library Loan and Document Delivery Benchmarking Study", *National Library of Australia*, 2001, Retrieved 1 Jul 2006 from http://www.nla.gov.au/initiatives/nrswg/illdd_rpt_sum.html

[16] R. Missingham, and M. Moreno, "Resource sharing in Australia", *Interlending & Document Supply*, Vol. 33 Issue 1, 2005, pp. 26-34.

[17] L. Porat, "Automation of inter-library loan services: effects on the patron and the library", *Interlending & Document Supply*, Vol. 29 Issue 3, 2001, pp. 108-113.

[18] J. Farrelly and D. Reid, "International perspectives in a New Zealand context", *Interlending & Document Supply*, Vol. 31 Issue 4, 2003, pp. 228-236.