IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 3, Ser. IV (May - June 2020), PP 62-67
www.iosrjournals.org

RMI Based Distributed Research Information Management and Usage: In case of Debre Markos University Burie Campus

Robel Mamo
Lecturer at Department of Computer Science, Burie Campus, Debre Markos University, Ethiopia

***Abstract*:** *The three main focuses of higher education institutions are teaching learning, research and community services. Conducting research to solve problems in different thematic areas is a vital for the development of once country. Supporting this vital activity using ICT is also very important for effective management and efficient usage of research information. In this study, object based Distributed Computing is used to develop RMI Based Distributed Research Information Management System (RDRIMS) for Debre Markos University Burie campus research and community service office (RCSO). Java Remote Method Invocation (RMI) is used to implement easily accessible, transparent, open and scalable distributed research information system that can improve the management and usage of research information on the campus. Moreover, the developed system is secured and easily maintainable.*

Key word: *Distributed Computing, RMI, RDRIMS, Debre Markos University Burie campus, RCSO*

Date of Submission: 16-06-2020 Date of Acceptance: 02-07-2020

I. Introduction

Higher Education institutions (HEIs) are mainly mandated to focus on teaching learning, Research and Community services. Research is a systematic way of collecting and analyzing data for seeking solution to problems, discovering new technologies and paving ways to development of a country [1]. Data are the lifeblood of today's organizations, and the effective and efficient management of data is considered an integral part of organizational strategy [2]. The traditional method of data storage has shown its impact on managing documents from security, retrieval, and monitoring [3]. For effective management and usage of research information a Research Information Management System (RIMS) is required and ICT is an emerging technology used to implement such kinds of modern data handling and information management systems. Developing RIMS using this emerging technology can change the culture of data handling and information usage in a secured, easily accessible, and suitable manner. An implementation of RIMS in universities or libraries ensures the proper management of research information for future use [4]. In HEIs RIMS promotes sharing and reuse of research works, provides necessary support documents, and communicate latest news and events to the researchers.

Debre Markos University Burie Campus Research and Community Service Office (RCSO) is coordinating and promoting research works in and outside of the campus. It is striving towards enhancing research culture, but lacks a modern research information management system. Currently, the office doesn't have any automated RIMS and hence it is doing its works manually. The manual work is creating so many challenges like accuracy, efficiency, reliability and security problems. It is revealed that the office requires a modern RIMS to alleviate these problems and provide a better service to the researchers. Therefore the aim of this study is to analyze, design and implement a RMI Based Distributed Research Information Management System (RDRIMS) as per the requirements of the office.

A distributed system is a collection of autonomous computers where interactions are done in a consistent and uniform way regardless of the difference in the type of computers and the network technology used to interconnect the computers. It used to implement networked applications in a client/server manner where clients are remote programs that request a service from the server and the server responds back to the clients. Remote Method Invocation (RMI) is used to connect components in object based architecture. Therefore, RDRIMS is a distributed system that uses Java RMI to connect the clients and sever transparently.

II. Statement of the Problem

The underlining problem that commences this study is the fact that manual handling of information is inaccurate and inefficient to manage and use research information. The major problems found in the existing manual work are as follows:

- Researchers are not able to submit their papers online.
- Researchers are not able to track their papers status online and on time.

- There is no online research papers archive.
- Researchers are not able to know easily which researches are conducted in the campus before.
- The office doesn't have a means of organizing research, researchers' and reviewers' information online.
- There is no automated means of generating research, researchers' and reviewers' report efficiently.
- There is no automated means of disseminating supportive documents to the researchers and reviewers.
- The process is time taking because it requires manually colleting, organizing and communicating research, researchers' and reviewers' information.
- Data is not confidential and unauthorized person may gain information about researches, researchers' and reviewers' information.

III. Main Objective of the Study

The main objective of the study is to analyze, design and implement RMI Based Distributed Research Information Management System for Debre Markos University Burie campus RCSO that can drastically improve the management and usage of research information.

IV. Related Literature

Distributed Computing

Distributed computing is a field of computer science that studies distributed systems [5]. Distributed computing refers to two or more computers networked together sharing the same computing work. The objective of distributed computing is to sharing the job between multiple computers [6]. A common misconception among people when discussing distributed systems is that it is just another name for a network of computers. However, this overlooks an important distinction. A distributed system is built on top of a network and tries to hide the existence of multiple autonomous computers [7].

Object Based Distributed Models

Distributed objects run in different process and use a remote interface for controlling access to its data and functionalities that can be accessed from other objects in other processes located on the same or different machines. Object based distributed models include CORBA (Common Object Request Broker Architecture), DCOM (Distributed Common Object Model) and Java RMI (Remote Method Invocation).

CORBA defined by the Object Management Group (OMG) is developed for communication of systems that are deployed on diverse platforms. CORBA is used for collaboration between systems on different operating systems, programming languages, and computing hardware [5].

DCOM is a set of Microsoft concepts and program interfaces in which client program objects can request services from server program objects on other computers in a network. DCOM is based on the Component Object Model (COM), which provides a set of interfaces allowing clients and servers to communicate within the same computer (that is running Windows 95 or a later version) [5].

RMI is a mechanism that allows one to invoke a method on an object that exists in another address space. The other address space could be on the same machine or a different one [5].

Java RMI (Remote Method Invocation)

The Java Remote Method Invocation Application Programming Interface (API), or Java RMI, is a Java application programming interface that performs the object-oriented equivalent of remote procedure calls (RPC) [8]. Java remote method invocation is an object oriented middleware system in which a user is able to call methods on object residing in another java virtual machine which could also be located in a remote machine [9]. RMI used to support object based distributed applications. The nature of objects that hides its data and functionality from the outside world by means of distributed interfaces enables the implementation of RMI to enhance distribution transparency.

V. Methodology

Design Methodology

The methodology used in this study is the Object-Oriented system development methodology. It is a software development strategy based on the idea of building systems from reusable objects. Objects are instances of classes and can be easily replaced, modified, and reused. It used to model real-world objects better than the traditional methods. It promotes reuse, modularity, extensibility, maintainability and improves productivity.

Method of Data collection

Gathering requirements is the corner stone to any successful study. A requirement is a statement that identifies a capability, characteristic or quality factor of a system in order for it to have value and utility to the user. Requirements are critical for scoping, defining, estimating, and managing the project. For effective elicitation of requirements, proper gathering techniques are required. The preferred methods of data collection used in this study are observation, interview, document analysis, and use case analysis.

- Observation is used to observe how the existing system works and provides service to the users.
- An Interview is used to gather the detailed requirements of the system. RCSO vice dean of the campus and researchers are selected for the interview.
- Document analysis is used to analyze the various documents (research guideline, reporting forms, document archives, etc) found in the office.
- Use case analysis is used to identify the scenarios, actors and use cases of the system.

Development tools

The system development tools used are Java Server Page (JSP) and RMI on the server side, HTML, CSS and Java Script on the client side, MySQL database server to create the database, GlassFish server to deploy the system, Anim-Fx to create animated effects and the Jaspersoft iReport tool to create dynamic reports to the system.

System Architecture

In this study Java RMI architecture is used to develop RDRIMS. RMI uses a request-reply communication to exchange messages between the client and server machines. An object's interface known as proxy is used to marshal remote method invocation into messages to be sent to the server and unpacks reply messages to return the result of the method invocation to the client. In the server side a stub, known as a skeleton is used to un-marshal request messages coming from the client and marshal reply messages to be sent to the client. When the clients invoke the remote methods it is transparent i.e. the clients don't know that the methods are executing on a different machine or vice versa.

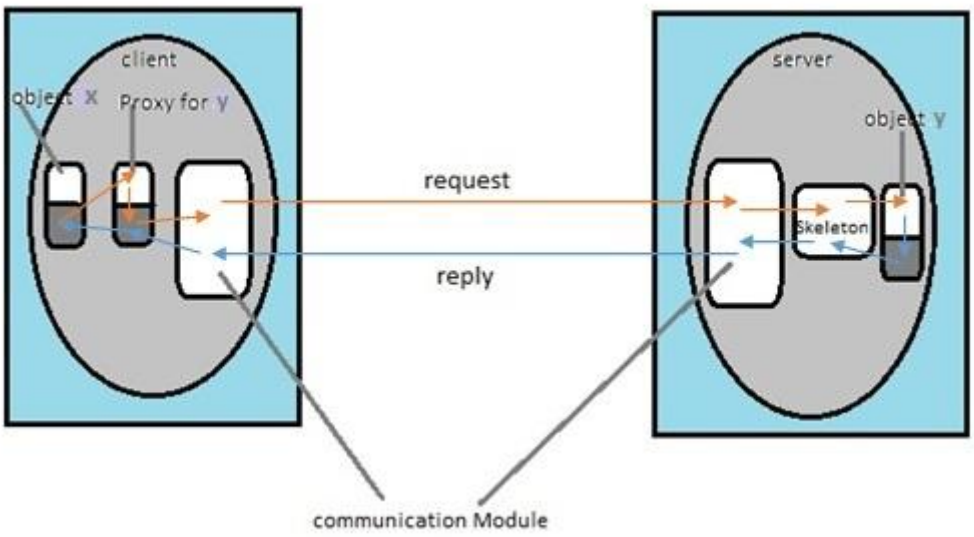

Figure 1. Request-reply view of Java RMI Architecture

VI. Implementation

Implementation of a distributed system using Java RMI requires programming the interface, implementation, server and the client. The interface program contains the declaration of remote method which is actually running on the server side, the implementation program contains the class that defines the method declared in the Interface program, the server program listens to Client requests and the client program invokes the remote method [10].

```java
package paperinterface;

import java.rmi.Remote;
import java.rmi.RemoteException;

/**
 *
 * @author Robel
 */
public interface newPaperInterface extends Remote {
    public int removePaper(String paperId) throws RemoteException;
}
```

Figure 2. Java RMI interface program

The figure above shows an interface called newPaperInterface which contains the declaration of removePaper method. The interface extends the java.rmi.Remote interface and the method throws RemoteException.

```java
package paperimplementation;

import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;
import paperinterface.newPaperInterface;
/**
 *
 * @author Robel
 */
public class PaperImplementation extends UnicastRemoteObject implements newPaperInterface {
 public PaperImplementation()throws RemoteException{
   }
 public String removePaper(String paperId) throws RemoteException {
    String message=null;
/*
try{
PreparedStatement pr=con.prepareStatement("SELECT * FROM paper WHERE PaperID=?");
pr.setString(1, paperId);
ResultSet rs=pr.executeQuery();
...
  */
     return message;
   }
}
```

Figure 3. Java RMI implementation program

The figure above shows the implementation program that defines the removePaper method declared in the new PaperInterface. It extends the java.rmi.server.UnicastRemoteObject and implements the newPaperInterface.

```java
package paperserver;

import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import paperimplementation.PaperImplementation;

/**
 *
 * @author Robel
 */
public class PaperServer {
    public static void main(String[] args) {
        try{
            Registry r=LocateRegistry.createRegistry(1099);
            PaperImplementation i=new PaperImplementation();
            r.bind("paper", i);
            System.out.println("Server started");
         }
        catch(Exception ex){
            ex.printStackTrace();
        }
    }
}
```

Figure 4. Java RMI server program

The figure above shows the server program which creates a registry using RMI dedicated port number 1099. The server program binds the remote object and waits for the client request.

```
<%@page import="paperinterface.newPaperInterface"%>
<%@page import="java.rmi.registry.Registry"%>
<%@page import="java.rmi.registry.LocateRegistry"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Remove Paper</title>
    </head>
    <body>
        <%
        String paperId=request.getParameter("paperId");
        try{
         Registry reg =LocateRegistry.getRegistry("127.0.0.1",1099);
         newPaperInterface pi =(newPaperInterface)reg.lookup("paper");
         out.println(pi.removePaper(paperId));
        }
         catch(Exception ex){
            ex.printStackTrace();
         }
        %>
    </body>
</html>
```

Figure 5. Java RMI client program

The figure above shows the web client program that invokes the removePaper remote method and displays the result of the remote method invocation.

VII. Advantages of the New System

The new system benefits researchers, reviewers and the RCSO by providing effective management and efficient usage of research information in a modern way. The main advantages of the new system are described below:

Accuracy

The Java RMI creates a proper communication between the clients and the server, data is properly abstracted and stored in the database, user inputs are properly validated and verified and system outputs are properly delivered. These lead the system to be accurate and perform what it promises.

Reliability

Any error happening when using the system is properly handled through a try catch block and hence there is no chance of happening catastrophic failure. Unless the user provides all the necessary input data there is no chance to perform an activity in the system and the system is not exposed to undesired and unseen alterations of data which are potential causes to failure.

Performance

The new system is developed using highly cohesive and loosely coupled classes and this will increase the modularity and decrease the complexity of the system which results performance enhancement. The Jaspersoft iReport tool also enables the system to generate dynamic reports efficiently.

Availability

The system is available in the intranet whether the internet is there or not on the campus. The system supports distributed location transparency which enables the clients to invoke the remote methods irrespective of their location. Generally the new system will make the research information available and easily accessible from its database through Java RMI technology.

Security

The system uses authentication mechanism to filter out valid users and their privilege to the system. In addition the MD5 hashing algorithm is used to secure login passwords. The users' session is well managed in the system to safeguard their session from others and the object based Java RMI promotes encapsulation and data hiding to protect object's data and functionality from other parts.

Maintainability

The system uses highly cohesive and loosely coupled classes and these make the system easily maintainable and scalable. Objects can be extended to add new data and functionalities easily when needed.

VIII. Conclusion

In this study, RMI based distributed research information management system is developed for Debre Markos University Burie campus research and community service office. The Java RMI object based distributed model is used to implement the system under the object-oriented system development approach. Highly cohesive and loosely coupled classes are used to increase the modularity and decrease the complexity of the system. The new system is easily accessible, transparent, open and scalable distributed system which guarantees effective management and efficient usage of research information on the campus. Moreover, the developed system is secured and easily maintainable. It is recommended that developing some of the functionalities of the system using a mobile (Android) application will able to increase the accessibility of the system.

References

[1]. Harmonized Research and Publication Guideline for the Forum of Higher Educational Institutions in the Amhara Region, 2018
[2]. Yaser Hasan Al-Mamary, Alina Shamsuddin, Nor Aziati. Management Information Systems,
a. Vol. 8 (2013), No. 4, pp. 010-017
[3]. Caluza, Las Johansen B. "Development of Electronic Document Archive Management System (EDAMS): A Case Study of a University Registrar in the Philippines." International Journal of Digital Information and Wireless Communications, vol. 7, no. 2, 2017, p. 106+. Accessed 7 Mar. 2020.
[4]. T R, Manu &Parmar, Minaxi&Aavarti, Shashikumara&Asjola, Viral. (2019). Research Information Management Systems: A Comparative Study. 10.4018/978-1-5225-8437-7.ch003.
[5]. Hanumant Pawar, Sujeet Patil, Sourabh Karche, Udit Upadhayay, Mahesh Channaram. Distributed Object Computing Using Java Remote Method Invocation, International Journal of Emerging Engineering Research and Technology Volume 3, Issue 5, May 2015, PP 68-76
[6]. Md. Firoj Ali, Rafiqul Zaman Khan. Distributed Computing: An Overview. Int. J. Advanced Networking and Applications, Volume: 07 Issue: 01 Pages: 2630-2635 (2015)
[7]. K Nadiminti, MD De Assuncao, R Buyya. Distributed Systems and Recent Innovations: Challenges and Benefits, InfoNet Magazine, 2006
[8]. Sutone, Saurabh & Mudliyar, Shivam & shukla, Rajat. NETWORK MONITORING USING RMI TECHNOLOGY, International Journal of Advanced Innovative Technology in Engineering (IJAITE), Vol. 1, Issue 2, March-2016
[9]. Eke, Christopher & Ezugwu, Absalom & Nweke, Henry & Aydin, Mehmet. (2013). Use of Java RMI on Mobile Devices for peer to peer computing. Journal of Network and complex systems. 3. 1-9.
[10]. Padole, Dr. Mamta. (2017). Big Data Analysis on Heterogeneous Distributed Systems using Remote Method Invocation. IOSR Journal of Computer Engineering. 19. 70-75. 10.9790/0661-1902057075.

Robel Mamo. "RMI Based Distributed Research Information Management and Usage: In case of Debre Markos University Burie Campus." *IOSR Journal of Computer Engineering (IOSR-JCE),* 22(3), 2020, pp. 62-67.