University of the Philippines Manila
College of Arts and Sciences
Department of Physical Sciences and Mathematics

CSCoRE: Computer Science Collaborative Research Environment 2.0

A special problem in partial fulfillment
of the requirements for the degree of
Bachelor of Science in Computer Science

Submitted by:

Vienna Blessilda V. Rom

April 2013

ACCEPTANCE SHEET

The Special Problem entitled "CSCoRE: Computer Science Collaborative Research Environment 2.0" prepared and submitted by Vienna Blessilda V. Rom in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Richard Bryann L. Chua, M.Sc.
Adviser

EXAMINERS:

	Approved	**Disapproved**
1. Gregorio B. Baes, Ph.D. (*candidate*)	_____	_____
2. Avegail D. Carpio, M.Sc.	_____	_____
3. Aldrich Colin K. Co, M.Sc. (*candidate*)	_____	_____
4. Perlita E. Gasmen, M.Sc. (*candidate*)	_____	_____
5. Ma. Sheila A. Magboo, M.Sc.	_____	_____
6. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
7. Geoffrey A. Solano, M.Sc.	_____	_____
8. Bernie B. Terrado, M.Sc. (*candidate*)	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

Avegail D. Carpio, M.Sc.
Unit Head
Mathematical and Computing Sciences Unit
Department of Physical Sciences and Mathematics

Marcelina B. Lirazan, Ph.D.
Chair
Department of Physical Sciences and Mathematics

Alex C. Gonzaga, Ph.D., Dr.Eng.
Dean
College of Arts and Sciences

Abstract

Collaboration is the cooperation of individuals in order to attain a common goal. In line with this concept, workflows may be employed to systematize certain procedures necessary to conduct a research or a project. Using workflows relative to collaboration yields increased efficiency, better process control, flexibility and process optimization. Meanwhile to adapt with the distributed model of organization of most software development teams, distributed version control systems (DVCSs) are rapidly gaining ground as the preferred architecture of projects over the traditional centralized systems. Offline operations, experimental branches and easy peer-to-peer collaboration are but some of the reasons why developers are making the switch.

Although CSCoRE integrates effective project management and availability of tools in one collaborative environment, it fairly lacks two substantial features relevant to research and development in Computer Science. First, it has no support for workflow management and second, it does not offer Git DVCS.

With the Computer Science Collaborative Research Environment (CSCoRE) 2.0, the collaboratory gains a step up on its purpose of being a beneficial tool for collaboration specifically for projects in the field of Computer Science through the addition of the modules ProcessMaker and Git repository. ProcessMaker provides CSCoRE with an embedded workflow management system in which pertinent workflows tailored to the needs of a project can be utilized to streamline the flow of information among members of the team. The Git repository on the other hand, offers users the ability to conveniently browse and obtain version controlled files of a project. Through this, the distributed nature of Git gives enough flexibility to fit well with the structure on how most software development teams collaborate and contribute.

Keywords: Collaboratory, Software Development System, ProcessMaker, Git

Contents

Acceptance Sheet **i**

Abstract **ii**

List of Figures **v**

I. Introduction **1**

A. Background of the Study 1

B. Statement of the Problem 2

C. Objectives of the Study 3

D. Significance of the Project 4

E. Scope and Limitations 5

II. Review of Related Literature **6**

III. Theoretical Framework **16**

A. Workflow and Business Process Management 16

B. Examples of Workflows for Software Development 17

C. ProcessMaker 18

D. Web Services 23

D..1 eXtensible Markup Language (XML) 24

D..2 Web Services Description Language (WSDL) 25

D..3 Simple Object Access Protocol (SOAP) 25

D..4 Universal Description Discovery and Integration (UDDI) . . 25

D..5 ProcessMaker WSDL Web Services 26

E. Git 27

IV. Design and Implementation **30**

A. Context Diagram 30

B. Use Case Diagram 30

B..1 Use ProcessMaker Workflow Management Module 31

B..2 Access Project 38

V. Architecture **42**

A. System Architecture 42

A..1 ProcessMaker Web Services 43

A..2 JGit 43

B. Technical Architecture 44

B..1 Server 44

B..2 Client 44

VI. Results **45**

A. ProcessMaker Module 45

B. Git Repository Module 52

VII. Discussions **55**

A. ProcessMaker Module 55

B. Git Repository Module 57

VIII. Conclusions **59**

IX. Recommendations **60**

X. Bibliography **61**

XI. Appendix **65**

A. Source Code 65

XII. Acknowledgement **66**

List of Figures

1 Relationship of Workflow and Business Process concepts 17

2 The nine core workflows in RUP 18

3 A Generic Software Development Workflow 19

4 A Generic Model for Reporting a Bug 19

5 A Generic Model for Fixing a Bug 20

6 ProcessMaker Architecture 21

7 ProcessMaker Object Model 22

8 Distributed Version Control diagram 28

9 Context diagram of CSCoRE 2.0 30

10 Top level use case diagram of CSCoRE 2.0 31

11 Top level use case diagram of ProcessMaker Workflow Management Module 32

12 Manage Process Users/Groups use case diagram 32

13 Assign User to Group activity diagram 33

14 Remove User from Group activity diagram 33

15 Manage Processes use case diagram 33

16 Import Process Definition activity diagram 34

17 Delete Process activity diagram 34

18 Initiate New Case activity diagram 35

19 Respond to Case use case diagram 35

20 Modify Draft Case activity diagram 36

21 Complete Task activity diagram 36

22 View Case use case diagram 37

23 View Case Information activity diagram 37

24 View Case Documents activity diagram 38

25 View Case by Category activity diagram 38

26 Access Project use case diagram 39

27 Access files in Git repository use case diagram 40

28 Browse Files and Branches use case diagram 40
29 Clone Repository use case diagram 40
30 View Change Log use case diagram 41
31 Pull from Remote Repository use case diagram 41
32 Push Changes use case diagram 41
33 CSCoRE system architecture 42
34 ProcessMaker Workspace Settings 45
35 ProcessMaker users page 46
36 Adding users to groups 46
37 Import a process 47
38 Delete a process 47
39 Initiate a new case 48
40 Respond to a case 48
41 View cases in inbox 49
42 View draft cases 49
43 View participated cases 50
44 View unassigned cases 50
45 View paused cases 51
46 View documents 51
47 Git repository settings 52
48 Browse Git repository 52
49 View a file 53
50 Git commit history 53
51 Downloading the Clone 53
52 Push changes to repository 54
53 Pull updates from remote repository 54
54 ProcessMaker Plugins Manager 56

I. Introduction

A. Background of the Study

Collaboration is the cooperation of individuals in order to attain a common goal [1]. Its application is gaining high regard from various people and institutions spanning different disciplines because of the several advantages it brings to a concerted effort. For one, it ensures effective use of individual talents. It allows the transfer of tacit knowledge or skills. It may also be a source of stimulation and creativity. It provides intellectual companionship. It extends a researcher's network through gaining contacts from other disciplines or institutions. Finally, it facilitates the dissemination of projects [2].

In line with collaboration, workflows may be employed to systematize certain procedures necessary to conduct a research or a project. A workflow is an automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules [3]. Using workflows relative to collaboration yields increased efficiency, better process control, flexibility and process optimization. Automation of processes results in the elimination of many unnecessary steps. Moreover, improved management of business processes is achieved through standardizing working methods and the availability of audit trails. Software control over processes also enables their redesign in line with changing business needs while focus on these processes leads to their streamlining and simplification [4]. Examples of varied systems that employ workflows are BioExtract Server[1], Microsoft Sharepoint[2] and Quark Dynamic Publishing[3], to name a few.

In the field of computer science, software development commonly demonstrates collaboration. The current situation of many software development teams in enterprises is characterized by having teams in different cities or countries, developers

[1]http://www.bioextract.org
[2]http://sharepoint.microsoft.com
[3]http://www.quark.com/Products/Quark_Publishing_System

working from home, and outsourced teams in separate locations [5]. Similarly, open-source software projects are made up of a large number of individuals or organizations who contribute development through a distributed fashion. With this they have managed to produce large, complex, and successful software systems such as the Apache Web server[4], Mozilla Web browser[5], and Linux kernel[6].

To adapt with the distributed model of organization, distributed version control systems (DVCSs) like Git are rapidly gaining ground as the preferred architecture of projects over the traditional centralized systems like Subversion (SVN) [6]. Offline operations, experimental branches and easy peer-to-peer collaboration are but some of the reasons why developers are making the switch [7].

The Computer Science Collaborative Research Environment (CSCoRE), a tool for collaborative software development and software project management specifically designed for Computer Science research, has been created by Jimenez as a step towards enhanced collaboration in the area. It provides a common interface where participants can create and join projects and have a single storage for data and information. CSCoRE enables the members of each project to manage and share data, references, tools and other relevant materials among themselves. The system also helps in maintaining current and historical versions of the files of the project with the use of SVN [8].

B. Statement of the Problem

Although CSCoRE integrates effective project management and availability of tools in one collaborative environment, it fairly lacks two substantial features relevant to research and development in Computer Science.

First, it has no support for workflow management. The absence of workflow automation typically results in skipping steps of a prescribed procedure in a project. Most workflow management systems can handle this but they are typi-

[4]http://www.apache.org
[5]http://www.mozilla.org
[6]http://www.linux.com

cally stand-alone applications which make use of their own database of users and files. Monitoring and synchronizing project resources with this independent software consequently becomes tiresome for the users. This unnecessary challenge of coordinating these applications often undermines the quality of project output.

Second, it does not offer Git distributed version control system (VCS). Git's momentum continues to grow in the open source community as more projects choose to use this system for source code management instead of SVN. SVN, which is included in CSCoRE, may have a fixed workflow model that is easy to use however it falls short of qualities like flexibility and speed that Git exhibits.

C. Objectives of the Study

To create the Computer Science Collaborative Research Environment (CSCoRE) 2.0 which will have the following two new modules:

1. ProcessMaker workflow management module which has the following functionalities for the respective CSCoRE roles:
    (a) Allow the project administrator to
        i. Assign members to groups in ProcessMaker
        ii. Import a ProcessMaker .pm file
        iii. Delete a process
    (b) Allow the project contributor, depending on the process permissions, to
        i. Initiate a new case
        ii. Respond to incoming cases
        iii. Modify Draft cases
        iv. View Participated cases
        v. View Unassigned cases
        vi. View Paused cases

vii. View Completed cases

viii. View Documents of a case

2. Git version control system module which has the following functionalities for the respective CSCoRE roles:

   (a) Allow the restricted user to

      i. Browse files and branches in the repository

      ii. Clone the repository

      iii. View repository change log

   (b) Allow the project contributor to

      i. Perform the functions of a restricted user

      ii. Pull changes from another remote repository

      iii. Push changes to the repository

D. Significance of the Project

Combining ProcessMaker and Git with the existing components of CSCoRE will leverage the efficiency in which the system can aid throughout the lifetime of a project.

With ProcessMaker, users will primarily be able to customize workflows to suit the processes for different research groups. Tasks will be organized and workload among team members can be properly distributed as necessary. Workflow automation can also ensure that procedures are strictly followed and can even be optimized to minimize time required to pass on a task from one person to another. Furthermore, prompt notifications and document sharing can improve communications between participants.

Meanwhile with Git, project teams can take advantage of being able to host their codes in a Git server and access them within the collaborative environment. The central repository will be the main hub for developers to acquire the latest

project files as well as browse the related branches. The contributing members of the project will also be aware of the changes applied to the repository contents.

E. Scope and Limitations

1. The project administrator shall import a ProcessMaker-generated .pm file containing the process definition.
2. Each task of the process defined in a .pm file must be assigned to a group.
3. Users can only be assigned to groups that are defined by the imported processes. If assigning a specific user directly to a task is desired, it must be done in the ProcessMaker application itself.
4. Users can only view and respond to cases of processes that have given their group the permissions to access, based on the process definition.
5. Documents of process cases are stored in the file system specified by ProcessMaker.
6. The efficiency of a process definition is not going to be evaluated by the module.
7. When a process has been deleted by the Project Administrator, he also has to manually delete the corresponding process in ProcessMaker.
8. The Git module shall be an interface for collaborators/integrators to manage the project repository.
9. Client software for Git must be installed by the members in order to perform Git local operations.
10. The central repository must be created and hosted in an external Git server.
11. Connection to a Git repository hosted on a website with a self-signed certificate using the HTTPS protocol will not work.

II. Review of Related Literature

Workflow management is a technology that is considered strategically important by many businesses, and its market growth shows no signs of abating. It can be used to help individuals manage their work and to provide a clear context for performing that work. A key challenge in the realization of this ideal is the reconciliation of workflow process models and software with the rich variety of activities and behaviors that comprise "real" work.

Abbott and Sarin deployed InConcert workflow management system in a variety of applications. On the positive side, processes moved faster (than before) because less time is wasted while important forms and documents are buried in piles on people's desks. The system demonstrated an open and extensible architecture by using industry-standard components such as a relational DBMS and offering an Application Programming Interface (API) that can be used to build custom workflow applications. On the other hand, the seemingly mundane issues of installation and administration, which arise with any software product, had become more visible and critical with a "glue" technology like workflow that needs to work with other applications and system software. Difficulty was also encountered in applying and matching the workflow process model to the way work is actually performed.[9]

A number of usability issues and challenges were then identified. One of them is that it is useful to distinguish between the procedural (structured aspects of the process) and nonprocedural content (unchoreographed interactions between people) of a work process. Interoperation between external activities (i.e. electronic mail), meetings and workflow (in both directions) is one kind of support that is needed, as is interoperation between disparate workflow systems. A mechanism (a generic capability or feature) and a policy (a prescribed use of the features) can be reconciled by providing sample policies that use the mechanisms in specific stylized ways which a customer can choose from and adapt as desired. The workflow system could also be further improved by generalizing the access control

model, specifically to allow privileges to be granted to a "role" in a process. Defining synchronization dependencies across hierarchy and process boundaries is also frequently necessary. Lastly, an evolutionary process must be developed with the characteristics of delayed binding, finding process definitions, process evolution and learning, and change control.[9]

Current workflow management technology offers rich support for process-oriented coordination of distributed teamwork. Reijers, et al. evaluated the performance of an industrial workflow process where similar tasks can be performed by various actors at many different locations. A large workflow process log with state-of-the-art mining tools associated with the ProM framework was analyzed. They concluded that there is a positive effect on process performance when workflow actors are geographically close to each other and that the use of workflow technology in itself is not sufficient to level geographical barriers between team members. Additional measures, such as assignment to favor the geographically close resource and caution on outsourcing, are required for a desirable performance.[10]

Georgakopoulos, et al. provided a high-level overview of the current workflow management methodologies and software products. In addition, distributed object management and customized transaction management were presented as infrastructure technologies that can address the limitations of current commercial workflow technology and extend the scope and mission of workflow management systems to support increased workflow automation in complex real-world environments involving heterogeneous, autonomous, and distributed information systems.[11]

Ying, et al. constructed a web services workflow based on XML Process Definition Language (XPDL). By virtue of two tools, workflow will enter the web stage, and theory is applied to concrete applications. Web services use a loosely coupled integration model to allow flexible integration of heterogeneous systems in a variety of domains including business-to-consumer, business-to-business and enterprise application integration. Since XPDL, a language standard that is used

to define workflow process, and web services are both based on XML standard, a strong workflow system in the web was constructed.[12]

Chan and Leung advocated the use of workflow systems to enact the process of software development. Besides being more general and flexible, the workflow paradigm supported useful features lacking in other approaches. Also, it helped reduce development complexity by allowing both the software development process and the software themselves to be captured using the very same paradigm. A workflow system was developed to support the software development process by presenting a solution to the ISPW-6 Software Process Example expressed in its specification language. They introduced a new and more general approach to software process enactment and identified new requirements for the workflow paradigm, such as event dependency, that are applicable to many other advanced applications.[13]

Mao and Zhang introduced workflow to describe the activities in the course of software testing. The process definition is viewed as a workflow engine to describe the control relation of each activity, and business form has also been discussed in detail as the information flow. The whole testing process has been divided into testing planning, testing design, testing execution and faults correction, and each sub-process is modeled through the typical workflow net. Finally, these four steps are all integrated into a complete testing process management framework, which can direct the construction of testing workflow management system.[14]

A process-oriented workflow system model which can support workflow configuration, process control, task assignment, task submitting and approval, and cooperative work was presented by Chen and Xin in [15]. The model is a hierarchical architecture that consists of three layers, i.e. database layer, workflow engine layer, and user interface layer. It is based on Browser/Server (synonymous to Client/Server) architecture which can provide multi-user, distributed and access control properties. The workflow engine is based on XPDL and relational database technologies and can support cooperative work, branch selection and

workflow configuration. The model was implemented on Windows .NET platform and applied in an international enterprise project management system. The application results demonstrated that this workflow system model can satisfy the requirement of cooperative project management.[15]

Sheth believes that today's workflow systems should evolve to what is termed as work coordination and collaboration systems (WCCSs). A WCCS will adapt to various changes the organization (including its interactions with external organizations) and the organizational processes being able to change the processes definitions as well as change the processes and component activities while they are being enacted or executed, and support a unified framework for managing coordination, collaboration, and information-based decision making activities that naturally occur as part of organizational processes. A WCCS can then not only support automation of the routine and well defined processes, but also support better human involvement and manage more complex, dynamic, and higher value mission critical processes. This can lead to significantly improved productivity and quality of results. It is further suggested that a multidisciplinary approach is an essential element of the part toward developing WCCSs.[16]

To deal with collaborative processes (choreographies), internal workflow models have to be aligned with the external behavior advertised through web service interfaces. However, traditional workflow management systems (WfMS) do not offer this functionality. Simply sharing and merging process models are often not possible, because workflow management lacks a widely accepted standard theory for workflow models. Multiple research and standardization efforts to integrate different workflow theories have been proposed over the years. XPDL is the most widely used standard for process model interchange and supported by over 80 systems. However, XPDL also lacks the possibility to relate a workflow model to its possible choreography interface abstractions. Haller, et al. remedied the situation by abstracting the XPDL model to a higher-level model, performing the integration and the compaction algorithms at that level and then grounding

it back to the desired choreography models. The procedure is adaptable to the target choreography model. The approach was demonstrated using Web Service Modelling Ontology (WSMO) choreographies which enables us to automatically generate interface models from any WfMSs that supports XPDL export.[17]

Pappas, et al. present the Patient Record Manager and the Workflow toolset of the wireless-based e-health system Ward-In-Hand, developed inside an IST European project and currently in use within three hospitals in Italy, Spain and Germany. The lack of homogeneity in the healthcare organizations required a suitable implementation of workflow automation tools to create and manage the execution of the caregiving processes, customizing them to local ward needs.[18]

Producers of business tools like Microsoft, Oracle, and SAP have also acknowledged the power of workflows to leverage the services they provide in varied aspects of a business like Enterprise Resource Planning (ERP), Enterprise Content Management (ECM), and product lifecycle management, to name a few.

Microsoft Sharepoint has the capability to enable collaboration amongst employees as they respond to the varying demands of customers and the market. However, there is a need for the platform to ensure documents and data are managed effectively. SharePoint's reliance on Windows Workflow Foundation (WF) as its process support mechanism severely constrains the platforms capabilities, inhibiting process architecture and leading to manually coded workarounds, which in turn, drive complexity and increased Total Cost of Ownership (TCO). Adding a comprehensive SharePoint-oriented Business Process Management (BPM) Suite to the mix solves many of the critical issues associated with widespread deployment of the platform.[19]

The Oracle E-Business Suite is an extensive set of business applications for managing important business processes across an organization. This suite of complex business applications requires extensive business and application process integration. Oracle Applications are designed to empower the business user to take accountability of the business process and resulting data. As a result, employees

using Oracle E-Business applications must perform a dual role of understanding the business process and the underlying supporting application tasks. Many times these application tasks are critical to the business process but are usually cumbersome and manual in nature. Manual tasks lend themselves to risk: data entry errors, errors in processing, and delays in processing. Additionally, when organizations change business processes or acquire new operating units, they are faced with complex change management issues from a business and IT perspective. UC4 Software offered an advanced solution for the Oracle E-Business Suite that streamlines business processes, reduces implementation time, and provides true enterprise process management within, around, and beyond the Oracle E-Business Suite.[20]

SAP Business Suite applications run customers' core mission-critical processes. Its approach to human tasks, for example, recognizes that users will continue to receive tasks and notifications directly from SAP Business Workflow, yet need a single point of access for task management, whether assigned from SAP Business Suite applications or new composite processes. SAP's go-to market approach is conservative by software industry standards, but appropriate in the context of core processes delivered globally across multiple industries and applications. The company relies on a long period of beta delivery and ramp-up before it declares new offerings like NetWeaver BPM to be generally available. SAP's BPM strategy embraces business-oriented standards like Business Process Modelling Notation (BPMN) - for execution as well as modeling - rather than hoping, as its chief competitors do, that the problems of roundtripping between BPMN and an IT-centric execution language will magically disappear. It also applies full-fledged business rules management within the BPM suite, including business-friendly decision table modeling. It also recognizes that if a new generation of BPM technology is really going to empower business, new skills and methodologies are needed in its customer base, and thus has embarked on a major effort in Business Process Expert (BPX) education, certification, and community support.[21]

ProcessMaker is an open source workflow management software application

that enables Small and Medium Enterprises (SMEs) to automate document intensive, approval-based processes across systems including finance, HR and operations. ProcessMaker is an AJAX enabled, Service-Oriented Architecture (SOA), web-based application that allows users across single and multiple sites to create and share workflows, customize forms, manage processes, and enhance reporting. It is the flagship product of Colosa, a developer of software solutions that enable companies to simplify workflow through the capture and automation of business processes. ProcessMaker, is one of the first applications to achieve certification through the Intel Certified Solutions Program and will be available on the Intel Business Exchange.[22]

De Una is a young company based in Quito, Equador that provides errand-running services complemented with messenger/courier service for home use and small business use. The web-based solution, developed by Ecuador-based IT consulting company Doknos, allows ProcessMaker to generate all the tasks related to an errand the moment that a client generates the request through a ProcessMaker web form. The process is then executed in ProcessMaker as tasks get passed from participant to participant. The application even integrates with mapping software to generate a logical order for those executing the errands based on geographic location of the particular errand. Through a special integration with a barcode reader system, Doknos is able to maintain a tight control over the status of all errands and the delivery of all courier packages. Doknos developed a plug-in to ProcessMaker using the advanced plug-in architecture offered by ProcessMaker. This system generates a bar code for each shipment allowing the company and customers to trace the process from beginning to end via the web and inform the customer through automatic alerts.[23]

Nuevatel SA, is one of the leading companies in the telecommunications sector in Bolivia, Haiti, and the Dominican Republic. The company identified the procurement process as one that had a high level of visibility within the company and would be an ideal starting point for a BPM implementation in ProcessMaker. Like

many procurement processes, their centralized process suffered from challenges like the physical travel of application forms for approval, lack of standardized format, and potential delays in the steps of the process. To improve the situation Nuevatel SA with the support of project team of Colosa Inc. implemented ProcessMaker with the following characteristics: system access for users in all departments; standardized application forms aligned with the company's ERP System - SAP Business One; single user authentication using Microsoft Active Directory (AD); process connection with SAP Business One to determine the valid parameters for goods or services and receive the status of the SAP purchase process; legal templates to create contracts that have been previously approved by the Legal Department; and automatic identification of levels of approvals needed and issues notifications to the users responsible at each point in the process. This resulted in the standardization of activities related to acquisitions, shorter and more efficient procurement times, complete traceability of every procurement process, replacement of paper forms with electronic forms, and clear path to automating additional company processes.[24]

EdAutomate focuses on taking the administrative out of the daily work so teachers and administrators can be focused on student achievement. With over 25 years in the educational software business, their team recognized an opportunity to work with school districts to automate these workflows by combining the best open source software into a suite and delivering the experience in a Software as a Service model (SaaS). Problems and challenges school districts encountered before automation include: too much time required in finding and filling out paper forms, forms data taking days to enter into data systems, or the forms data was never entered, resulting in loss of information available to manage the school district, and errors and lost documents resulting in additional lost time for all staff. The solutions provided by ProcessMaker are automated processes like EdBuyIt, EDHr, Maintenance and Technology Request, Reports and Analytics, Records Management, and Amazon EC2 Cloud Hosting. They were then able to achieve

more time on student achievement, increase transparency and have data driven decisions.[25]

With regard to version control systems (VCS), Git has numerous advantages over Subversion (SVN). First, the complete Git repository is stored locally in the clients system, making it extremely fast due to the absence of network latency; but with SVN it takes longer because all of the data is stored in a centralized repository. Git clients can commit changes to their localized repositories as new revisions while being offline whereas SVN users must be online in order to push to the repository from the working copy. There is also a smaller chance of data being lost in Git since data copies are stored locally in clients' systems; however with SVN, data loss in the central repository is permanent. The Git repository has efficient memory because the datas file format is compressed in addition to a small index file to store the information related to a particular file; while in SVN there are always two copies of a file in the working directory: one for actual work and the other for information to aid in operations. Furthermore, Git keeps track of contents while SVN keeps record of files. Most importantly, Git is based on the concept of branching that SVN does not have, so working directories of developers are simply different branches stemming from the same common base revision of the project. In Git a large number of users can commit or push data to the same repository because commits are not sequential unlike in SVN. Git also allows its users to have control over the merging of data in synchronized repositories. Although the facility to merge data is also there in SVN, it is limited because access controls are enforced across the entire project. [26]

Several statistics illustrate the extent of usage of Git compared to SVN. Based on Ohloh, an open source directory that features comprehensive metrics and analysis on thousands of open source projects, figures show that there has been huge growth on the use of both Git and SVN among recent years. At the time of writing, there are 268, 420 SVN repositories (54%) and 139, 677 Git repositories (28%) of all the open source repositories registered on the site [27] compared to 116,224

Git repositories (35%) and 145,917 SVN repositories (44%) in May 2011. Results from the Eclipse Community Survey for 2012 similarly show that Git definitely has the momentum in the source code management market with Git/Github usage increasing from 13% (2011) to 27% (2012) while Subversion continues to decline but is still the most popular [6]. In a similar vein, a 2009 GNOME DVCS Survey of SVN users, which intended to collect data related to a possible move for the GNOME project from SVN to a DVCS, found out that there is a strong preference in the community toward switching, and that Git has a strong lead followed by SVN, Bazaar, and Mercurial. Moreover, people familiar with or are regularly using a certain system tended to prefer SVN but Git always came in a positive second [28].

III. Theoretical Framework

A. Workflow and Business Process Management

Workflow and business process are two inter-related concepts.[29] Figure 1 relates some important terminologies regarding the two ideas.

A workflow is the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve or contribute to an overall business goal. It is a collection of procedures amenable to automation where the participants consist of humans or software applications.

A business process is a set of linked procedures or activities which collectively realize a business objective or policy goal, within the context of an organization structure defining functional roles and relationships. Workflow is a part of a business process, which involves numerous activities and resources.

Workflow management system (WfMS) is a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, able to interpret process definitions, interact with workflow participants and invoke IT tools and applications. It supports the definition, creation, execution and general management of workflows. It may execute on one or more Workflow Engines and it specifies the who, what and when of a process execution.

Business process management (BPM) entails the assessment, analysis, modeling, definition and subsequent operational implementation of the core business processes of an organization.

Business Process Management activities can be grouped into:

1. Design - capturing and defining business processes
2. Execution - representation and execution of processes
3. Monitoring - tracking and reporting the execution of processes

Business process management incorporates workflow management. Workflow

management is largely concerned with definition of processes and enactment services. On the other hand, BPM activities allow: process design before implementation by a workflow system; and monitoring, evaluation and re-design after implementation and during execution by workflow system.

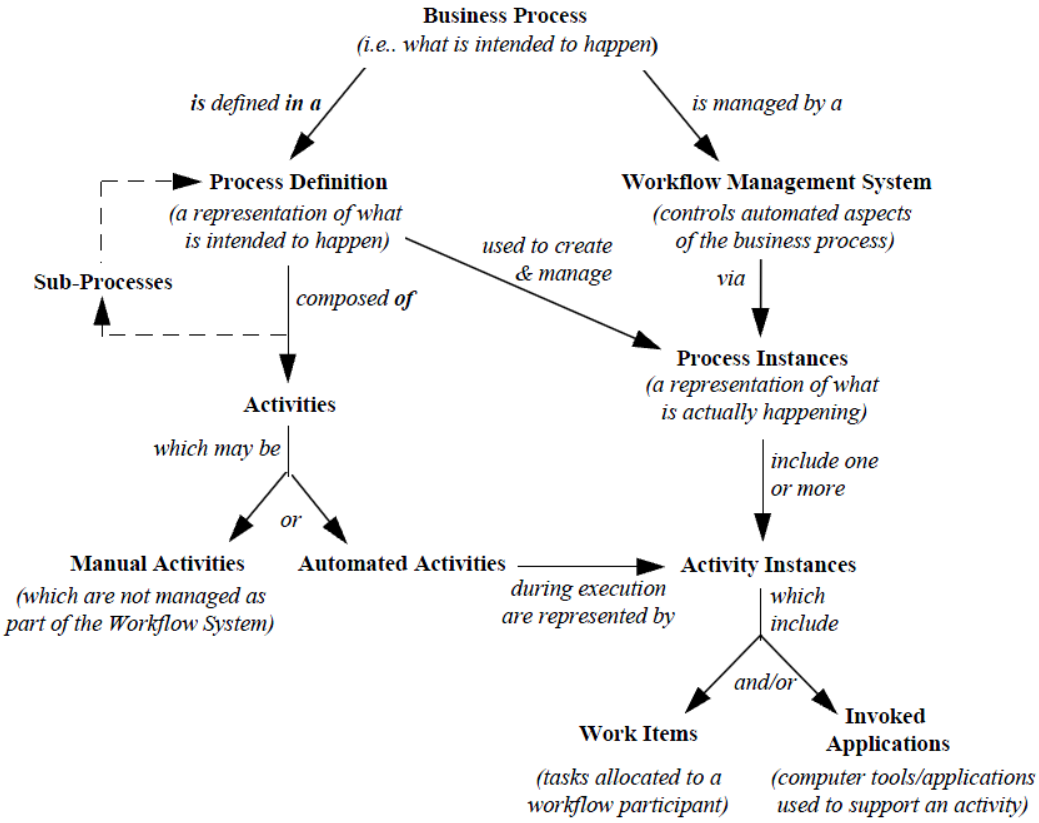


Figure 1: Relationship of Workflow and Business Process concepts

B. Examples of Workflows for Software Development

The Rational Unified Process (RUP) contains nine core process workflows (Figure 2) that represent a partitioning of all workers and activities into logical groupings.[30]

The core process workflows are divided into six core "engineering" workflows (business modeling, requirements, analysis & design, implementation, test and deployment) and three core "supporting" workflows (project management, configuration & change management and environment).

The actual complete workflow of a project interleaves these nine core workflows, and repeats them with various emphasis and intensity at each iteration throughout

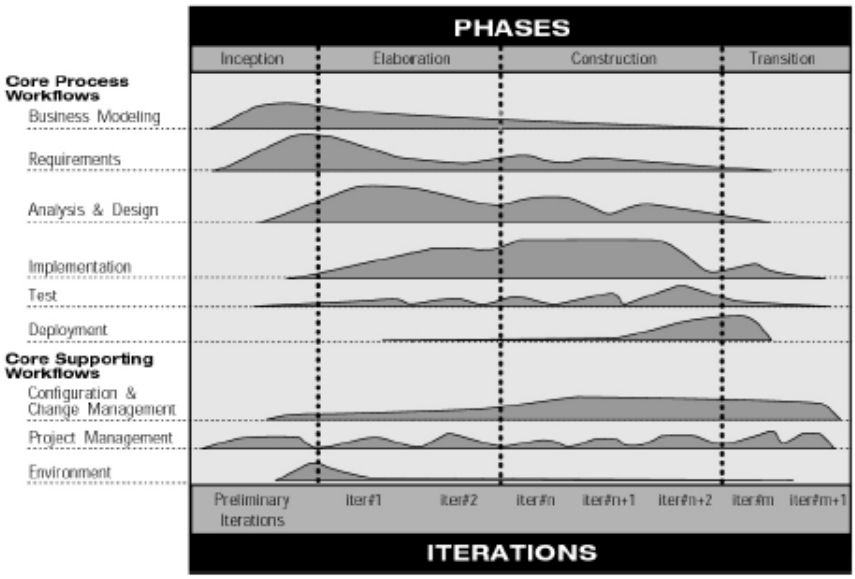


Figure 2: The nine core workflows in RUP

the lifecycle.

Figure 3 is an example of a generic software development workflow typically employed. [31] This software engineering process emphasizes its role- and purpose-oriented human aspects: who is involved, what are stakeholders' roles in the process's success, what responsibilities do they hold with respect to the system and what goals are they expected to pursue.

A specific example of a workflow is bug tracking, which can be viewed as one kind of problem resolution process. The software engineering community has established standards for such processes, as exemplified in ISO/IEC 12207. Figures 4 and 5 illustrate how to model two typical bug-tracking steps. A formal model can help developers visualize their process, remind them of their obligations and also allow process analysts to compare different models to each other: process vs. practice models, etc.[32]

C. ProcessMaker

ProcessMaker is an open source business process management (BPM) or workflow software application. It allows for the design, automation and deployment of any kind of business processes or workflows of an organization. This web based

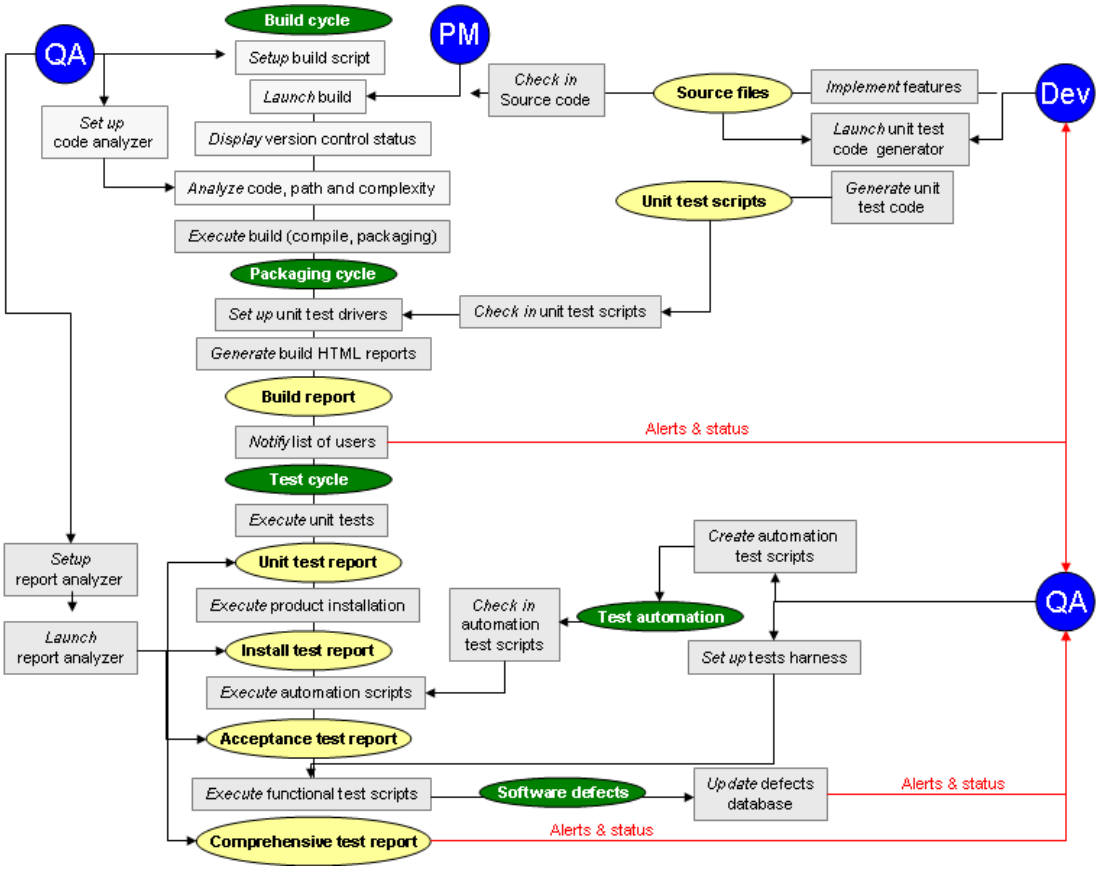
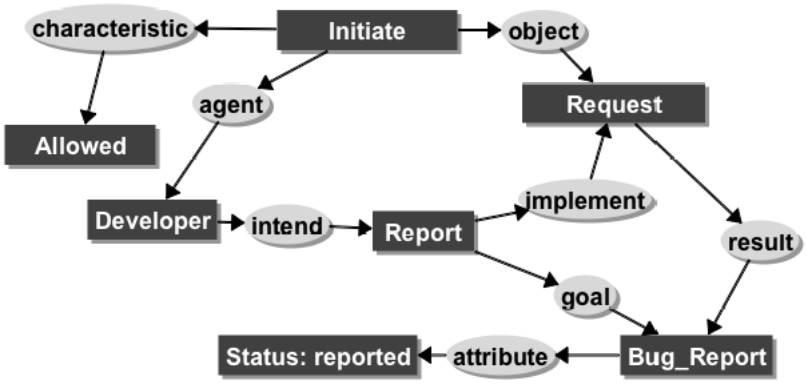


Figure 3: A Generic Software Development Workflow



There is a developer who is allowed to initiate a request whose result is a bug report with status "reported". The same developer intends to report the bug report using the request.

Figure 4: A Generic Model for Reporting a Bug

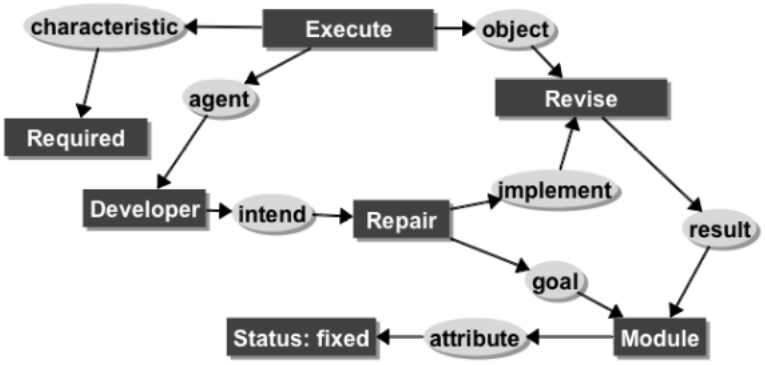


Figure 5: A Generic Model for Fixing a Bug

workflow software can integrate with other applications and systems such as ERP, business intelligence, CRM and document management.[33]

ProcessMaker is written in the PHP programming language and built using the Gulliver Framework, an open source framework for developing web applications and web services based upon a Model View Controller (MVC) model and Role Based Access Control (RBAC). It uses the PEAR DB classes abstracted behind a Gulliver class. ProcessMaker also uses Smarty, a template engine for PHP which facilitates a manageable way to separate application logic and content from its presentation.[34]

Figure 6 shows the relationship between the major components of the ProcessMaker Server and its interfaces with the outside world. ProcessMaker is designed on the LAMP/WAMP stack: Linux, UNIX or Microsoft Windows is the operating system; Apache, the web server; MySQL, the database server; and PHP, the programming language. It uses object-relational mapping software Propel to map between ProcessMaker's PHP classes and databases. Propel opens access to different DBMS, including MySQL, PostgreSQL, Oracle, and SQL Server (and Sybase if using Linux/UNIX). ProcessMaker is designed on the Gulliver framework, an open source development framework created by Colosa. It uses RBAC to manage user roles. It also uses PHP SOAP to manage web services with SOAP. ProcessMaker has two mail engines: PHP's built-in mail() function and PHP Mailer. Finally, it

is web-based and cross-browser, though it is optimized for Mozilla Firefox.

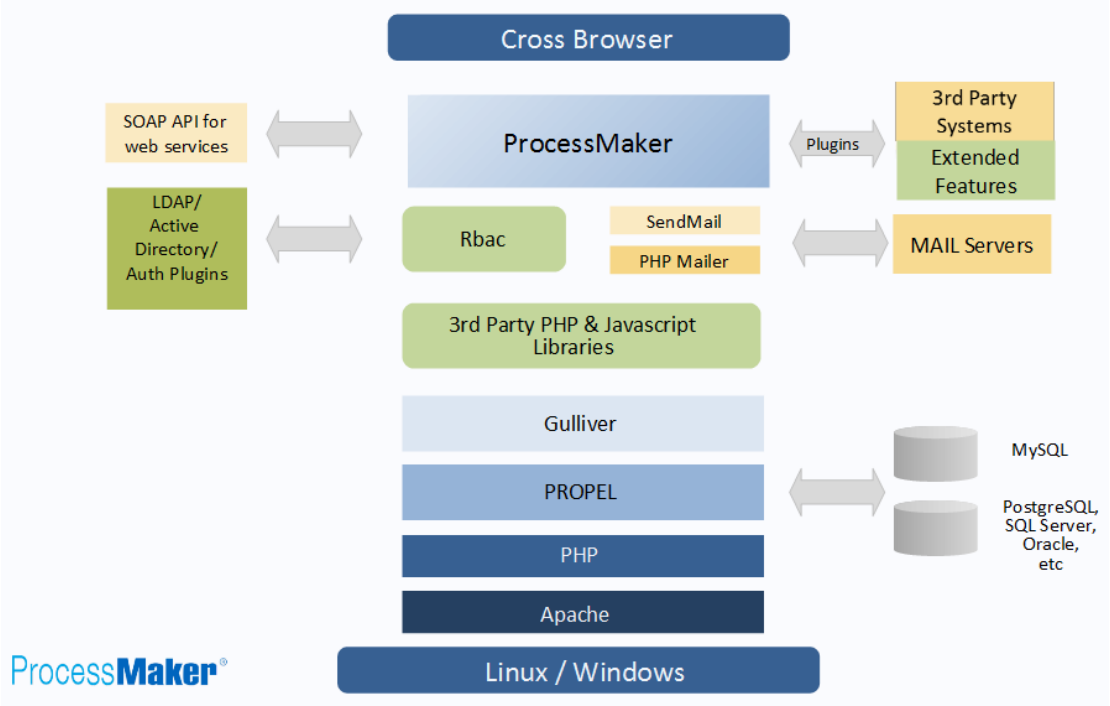


Figure 6: ProcessMaker Architecture

Using Simple Object Access Protocol (SOAP), ProcessMaker can connect through web services to other systems, including but not limited to DMS and CRM systems, middleware, messaging, PM Mobile, etc. Using Lightweight Directory Access Protocol (LDAP), ProcessMaker will be able to manage high user authentication.

ProcessMaker contains two main components - a design environment and a run-time engine. The design environment includes tools to map processes, define business rules, create dynamic forms, and add input and output documents. The run-time engine allows for cases to be started and run through the process. This engine turns the process map design into a fully-functioning application.

ProcessMaker organizes system users into users, groups, departments and roles.[35] A user account in ProcessMaker represents a person in an organization or an account with special privileges, such as the Administrator account. Users can be assigned to groups to simplify the assignment of tasks. Departments are another way to organize users based on an organizational structure. Each

user is assigned a role, which determines what actions that user can perform in ProcessMaker and what parts of the interface that user can access.

ProcessMaker follows an object model for process definition shown in Figure 7. [36]

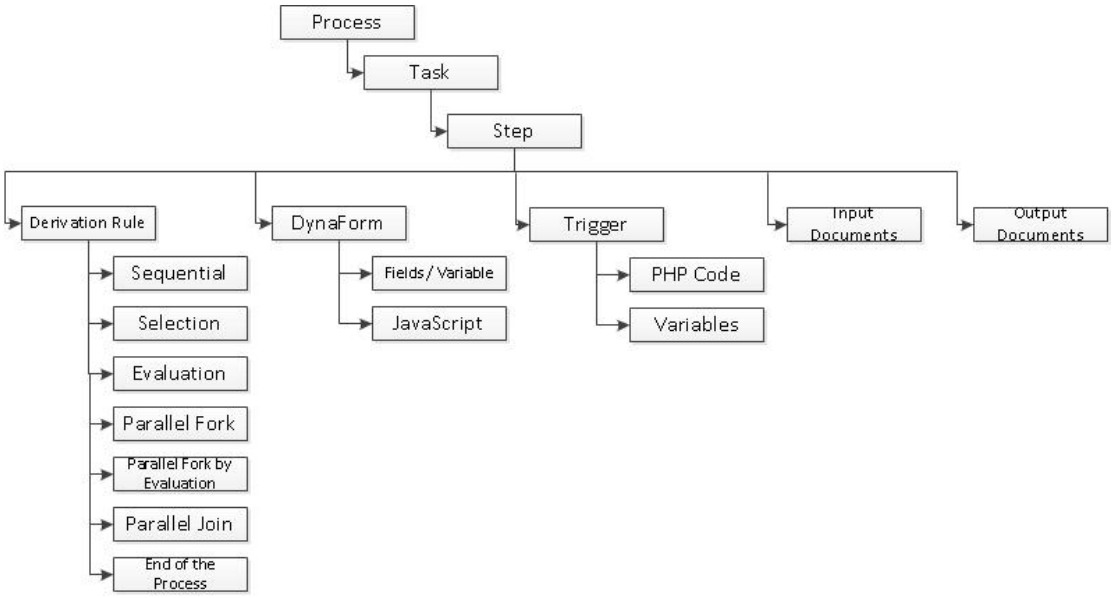


Figure 7: ProcessMaker Object Model

The process object gives the definition for its instances known as business cases. A task, which corresponds to a rectangle in a process map, defines the interaction of every object residing inside the business process definition. A step can be composed of a sequence of DynaForms, triggers, input documents and output documents. The derivation rule step is always executed upon task completion. The DynaForm allows a direct human-workflow interaction by means of a distributed web application approach. A trigger is a piece of code intended to express the most complex business rules by means of a high level programming language. Finally, input documents are those uploaded by the user while output documents are those generated by the system.

Processes are designed on the process map – a graphical user interface which facilitates the visualization of the process elements. An existing process may also be imported and adapted to the needs of the organization.

Processes designed in one installation of ProcessMaker can be imported into

another. This procedure will only import the process definition (including DynaForms, Input and Output Documents and triggers) and group accounts, but will not import user accounts, roles, or any cases. Two supported file types for import and export are the ProcessMaker process file with the .pm extension and the XPDL file with the .xpdl extension. As of the writing of this document, XPDL support in ProcessMaker is still in the beta version.

A case, or instance of a process, may be initiated by a user assigned to the first task of a process. Once a user is designated to work on a task in a case, it will appear in his Inbox.

The cases of a user are classified by type. The Inbox includes the list of cases that require input from a particular user. Draft cases are those that are being edited or advanced by the user but have not yet been submitted to the next task. Participated cases are those that the user has sent or participated in. Unassigned cases are those that allow any user from the pool of assigned users to grab the case and work on the task. Paused cases are those that for some reason have been temporarily stopped at this task, and are no longer advancing. For users with the "PM_SUPERVISOR" permission, cases for review are those that have been sent back to the supervisor. Lastly, for users with the "PM_REASSIGNCASE" permission, cases for reassignment are those that may be reassigned to another user.

When running a case, a user has access and can complete the necessary steps (such as DynaForms, Output and Input Documents) that are assigned to the current task. Information regarding the current process is also available. This includes the process map, process information, task information, case history, message history, DynaForms, uploaded documents and generated documents.

D. Web Services

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-

processable format (specifically WSDL). Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. [37]

Web services are powered by XML (eXtensible Markup Language) and three other core technologies: WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol), and UDDI (Universal Description Discovery and Integration). Before building a Web service, its developers create its definition in the form of a WSDL document that describes the service's location on the Web and the functionality the service provides. Information about the service may then be entered in a UDDI registry, which allows Web service consumers to search for and locate the services they need. Based on information in the UDDI registry, the Web services client developer uses instructions in the WSDL to construct SOAP messages for exchanging data with the service over HTTP. More about these core technologies is detailed below. [38]

D..1 eXtensible Markup Language (XML)

XML is a W3C (World Wide Web Consortium) specification that defines a meta-language for describing data. In XML applications, data is described by surrounding it with customizable, text-based tags that give information about the data itself as well as its hierarchical structure.

Because XML syntax consists of text-based mark-up that describes the data being tagged, it is both application-independent and human readable. This simplicity and interoperability have helped XML achieve widespread acceptance and adoption as the standard for exchanging information between heterogeneous systems in a wide variety of applications, including Web services.

XML forms the basis for all modern Web services, which use XML-based technologies to describe their interfaces and to encode their messages. WSDL, SOAP, and UDDI all use XML-based messaging that any machine can interpret.

D..2 Web Services Description Language (WSDL)

Also maintained by the W3C, WSDL is an XML-based format for describing Web services. Clients wishing to access a Web service can read and interpret its WSDL file to learn about the location of the service and its available operations. In this way, the WSDL definition acts as the initial Web service interface, providing clients with all the information they need to interact with the service in a standards-based way. Through the WSDL, a Web services client learns where a service can be accessed, what operations the service performs, the communication protocols the service supports, and the correct format for sending messages to the service.

The WSDL document that describes a Web service acts as a contract between Web service client and server. By adhering to this contact the service provider and consumer are able to exchange data in a standard way, regardless of the underlying platforms and applications on which they are operating.

D..3 Simple Object Access Protocol (SOAP)

SOAP is an XML-based protocol from the W3C for exchanging data over HTTP. It provides a simple, standards-based method for sending XML messages between applications. Web services use SOAP to send messages between a service and its client(s). Because HTTP is supported by all Web servers and browsers, SOAP messages can be sent between applications regardless of their platform or programming language. This gives Web services their characteristic interoperability.

Data is sent between the client(s) and the Web service using request and response SOAP messages, the format for which is specified in the WSDL definition. Because the client and server adhere to the WSDL contract when creating SOAP messages, the messages are guaranteed to be compatible.

D..4 Universal Description Discovery and Integration (UDDI)

UDDI is a standard sponsored by OASIS (Organization for the Advancement of Structured Information Standards). Often described as the yellow pages of Web

services, UDDI is a specification for creating an XML-based registry that lists information about businesses and the Web services they offer. UDDI provides businesses a uniform way of listing their services and discovering services offered by other organizations. Though implementations vary, UDDI often describes services using WSDL and communicates via SOAP messaging. Registering a Web service in a UDDI registry is an optional step, and UDDI registries can be public or private (i.e. isolated behind a corporate firewall). To search for a Web service, a developer can query a UDDI registry to obtain the WSDL for the service he/she wishes to utilize. Developers can also design their Web services clients to receive automatic updates about any changes to a service from the UDDI registry.

D..5 ProcessMaker WSDL Web Services

The ProcessMaker Web Services API is a set of SOAP Web Services which make it easy to embed ProcessMaker functions into 3rd party software or connect ProcessMaker to 3rd party systems. It provides a programming interface to remotely control and integrate ProcessMaker with external systems using Service-Oriented Architecture (SOA) compliant protocols. WSDL web services allow ProcessMaker's functionality to be accessed remotely over the internet using the SOAP 1.2 and WSDL 1.1. They can be called both in triggers and in external applications and scripts.

WSDL web services allow external programs to remotely retrieve lists of users, groups and roles, start and derivate cases and send notifications. These web services can be very useful when creating scripts to automate tasks in the server and building web pages which create and derivate ProcessMaker cases.

The SOAP and WSDL protocols were not designed for passing files over the internet, so ProcessMaker WSDL web services cannot be used to directly upload and download files such as Input Documents, Output Documents and Attached Files. Instead, ProcessMaker WSDL web services can be used to obtain the necessary information, so that other internet protocols can be used to upload and

download files to/from a ProcessMaker server.

E. Git

Git is a fast, open source, distributed version control system that is quickly replacing Subversion in open source and corporate programming communities. It was initially designed and created by Linus Torvalds for Linux kernel development. Git has been active for several years since 2005 and is written mostly in C. The full source code is available for anyone to analyze or improve upon. Git is licensed under the GNU General Public License.[39]

Git is a version control system (VCS), a simple command line tool for keeping a history on the state of a source code project. It allows the user to specify which files to track and periodically commit or save the state of the project at a particular point. Developers can then share that history for collaboration, merge work between one another, and compare or revert to previous versions of the project or individual files.

Git is fully distributed, which means that it can work almost entirely offline. In stark contrast to VCS tools like Perforce or Subversion, Git does nearly all of its operations without needing a network connection, including history viewing, difference viewing and committing. This also means that Git is very fast compared to those systems partially due to the fact that none of these operations has any dependency on network latency.

Git is also very space efficient. For example, comparison of the checkout/clone sizes of Subversion and Git shows that Git comes out very favorably, which is impressive considering that the Git clone contains the entire history of the project - every version of every file back to the first commit, whereas the Subversion checkout is just the last version of the project.

Git handles data as snapshots, not changesets. That is, instead of thinking about and storing commit points as file-based patches or changes, it stores it as a simple snapshot of what your project looked like when you committed. Commits,

then, are simply objects that contain some metadata about the commit (the message, author, date, etc.), a pointer to a single snapshot of the project and pointers to the commit that came directly before it. The commit history and data model of Git is really just a directed graph - a simple series of snapshots.

The most compelling feature of Git may be its branching model. Instead of the popular VCS branching method of simply cloning into a separate directory for a branch, Git lets a user switch between branches in a single working directory. Add to that the fact that creating and switching between branches is nearly instant, not all of your branches need to be shared, and it is easy to stash partially completed work - means that it can fundamentally change the way that a developer works. The model enables and encourages a non-linear development cycle, where a user can work on multiple lines of thought in parallel without them stepping on each other.

In Distributed Version Control Systems (DVCS) such as Git, clients do not just check out the latest snapshot of the files: they fully mirror the repository, as illustrated in Figure 8. Thus if any server dies, and these systems were collaborating via it, any of the client repositories can be copied back up to the server to restore it. Every checkout is really a full backup of all the data.

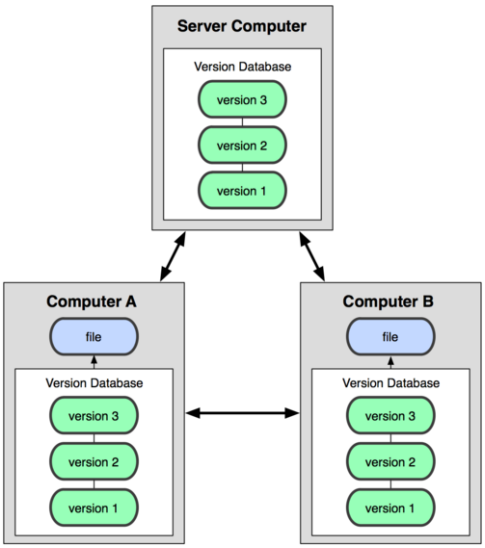


Figure 8: Distributed Version Control diagram

Unlike Centralized Version Control Systems (CVCSs), the distributed nature

of Git allows you to be far more flexible in how developers collaborate on projects. In Git, every developer is potentially both a node and a hub – that is, every developer can both contribute code to other repositories and maintain a public repository on which others can base their work and which they can contribute to. This allows you to set up several types of workflows that are not possible in centralized systems, such as hierarchical models.[40]

IV. Design and Implementation

A. Context Diagram

The system will have 6 access levels: system administrator, project administrator, project contributor, restricted user, registered user and non-registered user. The context diagram is shown in Figure 9.

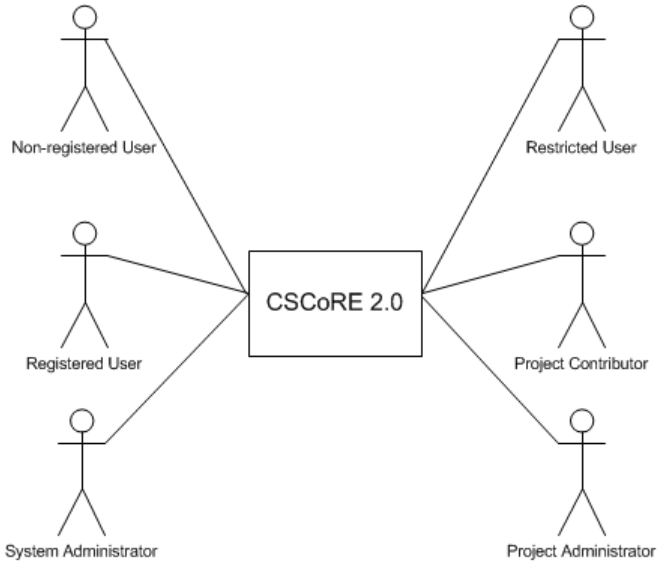


Figure 9: Context diagram of CSCoRE 2.0

B. Use Case Diagram

An non-registered user will be able to browse the list of projects and request for an account in the system. Upon approval of the account request, he will become a registered user. Aside from the capability to browse the list of projects, he will also be able to request for membership in a project. He will also be able to request for project creation. When the request is accepted, he will automatically become the administrator for that project. The project administrator will have the responsibility to manage the project and its members. The project members will have two types: the restricted users and project contributors. Both will have the access to the project but of different levels. The project administrator and contributors will also be able to use the ProcessMaker workflow management

module. Lastly, the system administrator will be responsible for the management of the whole system. Figure 10 shows the top level use case diagram of CSCoRE 2.0.

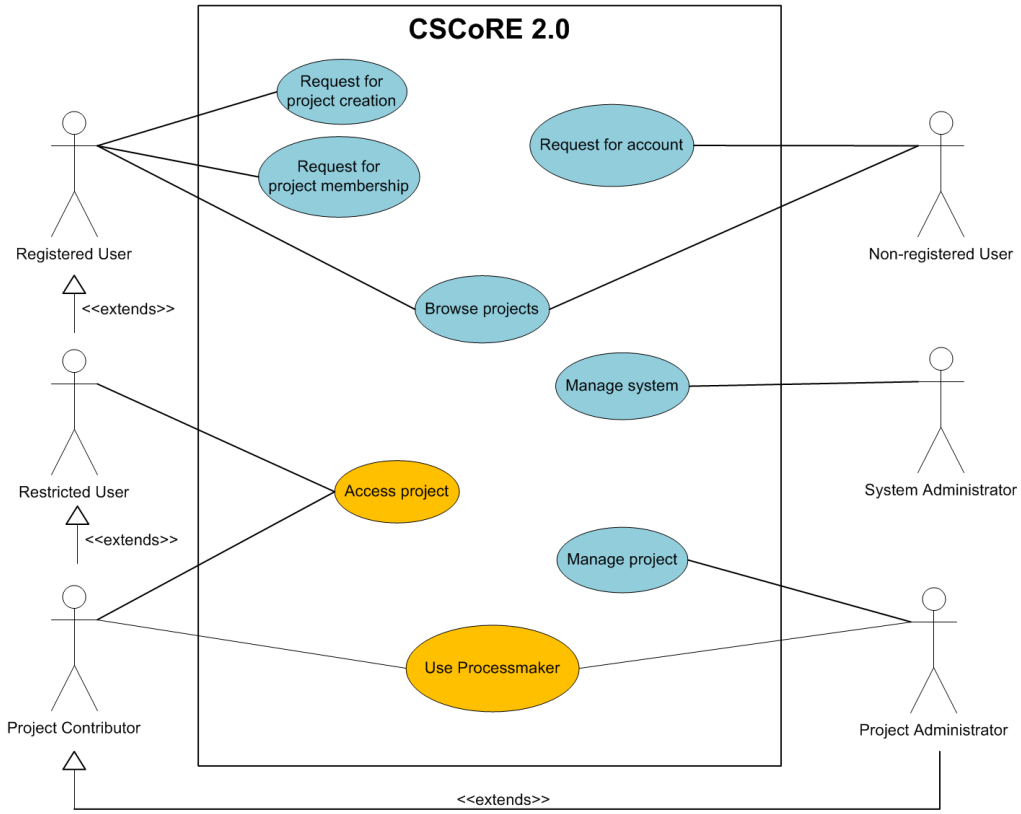


Figure 10: Top level use case diagram of CSCoRE 2.0

B..1 Use ProcessMaker Workflow Management Module

The project administrator has the resposibility to manage the processes of the project and the association of users to groups for each process. On the other hand, a project contributor may initiate a new case for a process that he is permitted to use. He may also respond to a case assigned to him and view cases that he has been a part of. All of these actions are completed with the ProcessMaker application. Figure 11 shows the use case diagram of the ProcessMaker Workflow Management Module.

B..1.1 Manage Process Users/Groups The project administrator can assign a user to a group required by a process. He can also remove a user from a

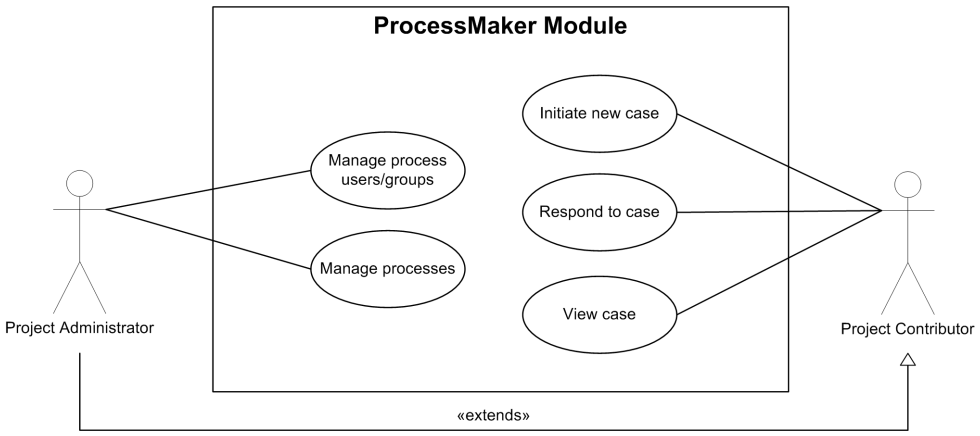


Figure 11: Top level use case diagram of ProcessMaker Workflow Management Module

group if necessary. Figure 12 shows the Manage Process Users/Groups use case diagram.

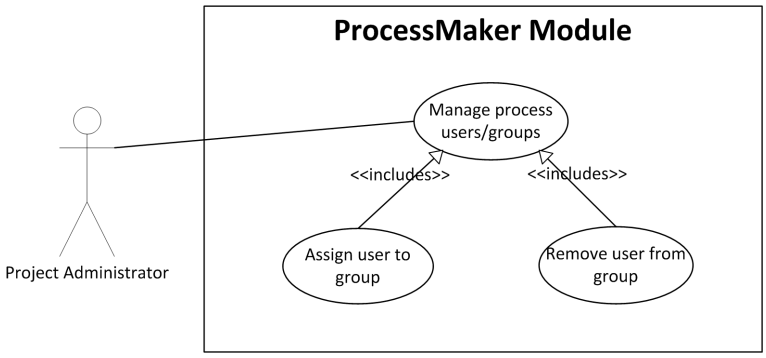


Figure 12: Manage Process Users/Groups use case diagram

B..1.1.1 Assign User to Group The project administrator can assign a user to a group to participate in a specific process. Figure 13 shows the Assign User to Group activity diagram.

B..1.1.2 Remove User from Group The project administrator can remove a user from a group of a specific process. Figure 14 shows the Remove User from Group activity diagram.

B..1.2 Manage Processes The project administrator can import a process definition for the use of his project. He may also delete a process in the project. Figure 15 shows the Manage Processes use case diagram.

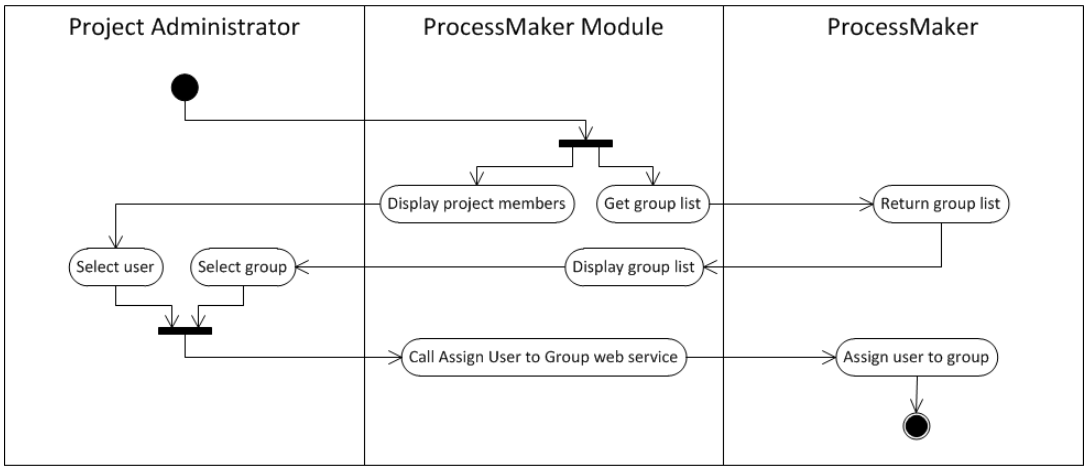


Figure 13: Assign User to Group activity diagram

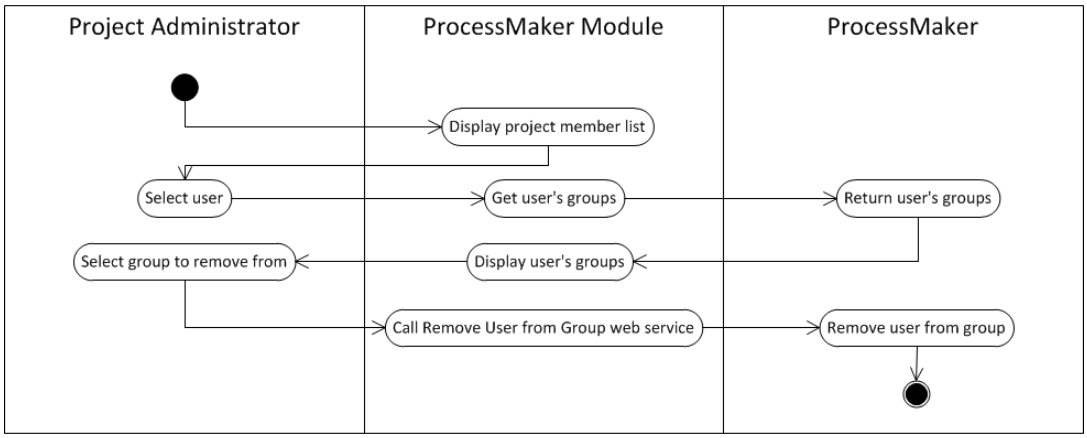


Figure 14: Remove User from Group activity diagram

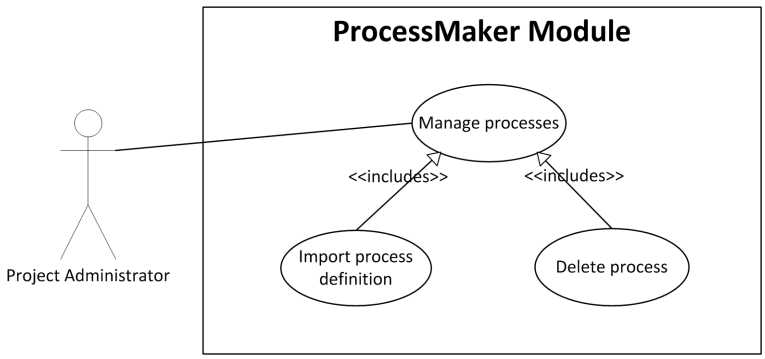


Figure 15: Manage Processes use case diagram

B..1.2.1 Import Process Definition The project administrator can import a .pm file which contains a process definition. Figure 16 shows the Import Workflow Definition activity diagram.

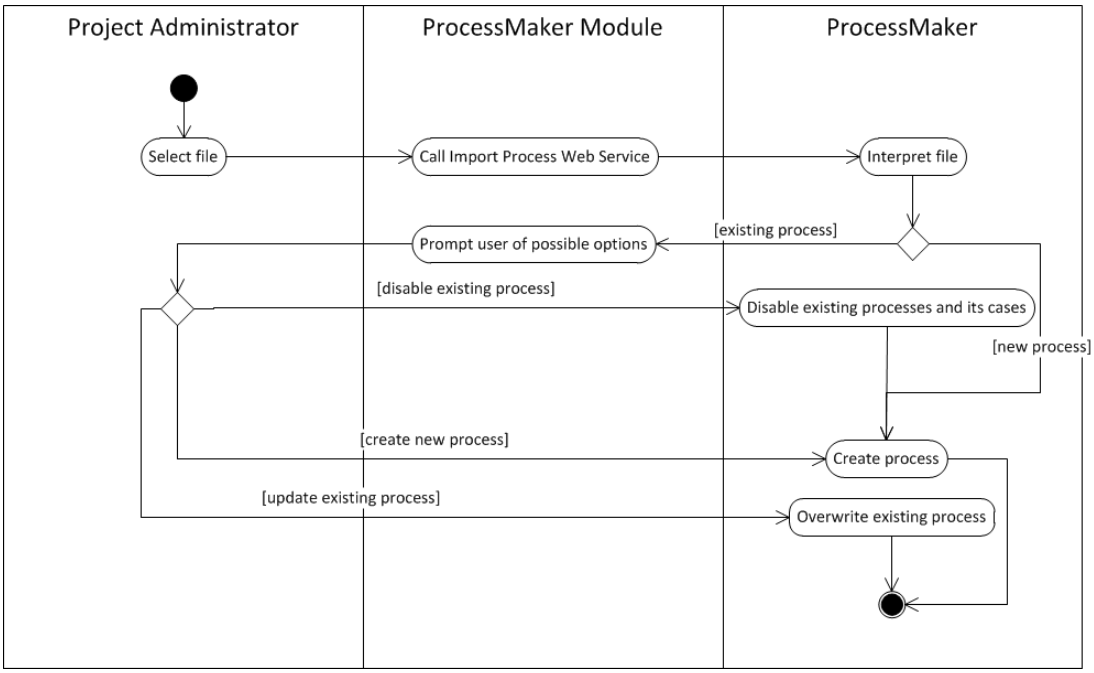


Figure 16: Import Process Definition activity diagram

B..1.2.2 Delete Process The project administrator can delete a process in the project. Figure 17 shows the Delete Process activity diagram.

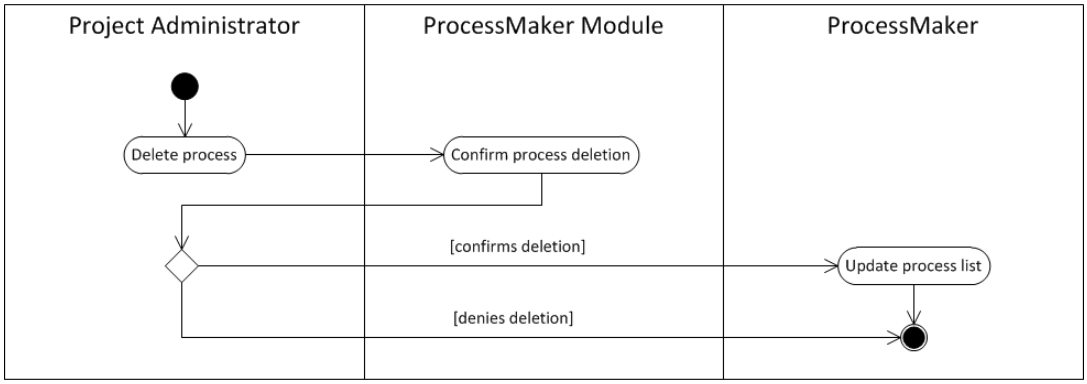


Figure 17: Delete Process activity diagram

B..1.3 Initiate New Case A project contributor can initiate a new case for a process in which he is a member of a group assigned to its initial task. Figure 18 shows the Initiate New Case activity diagram.

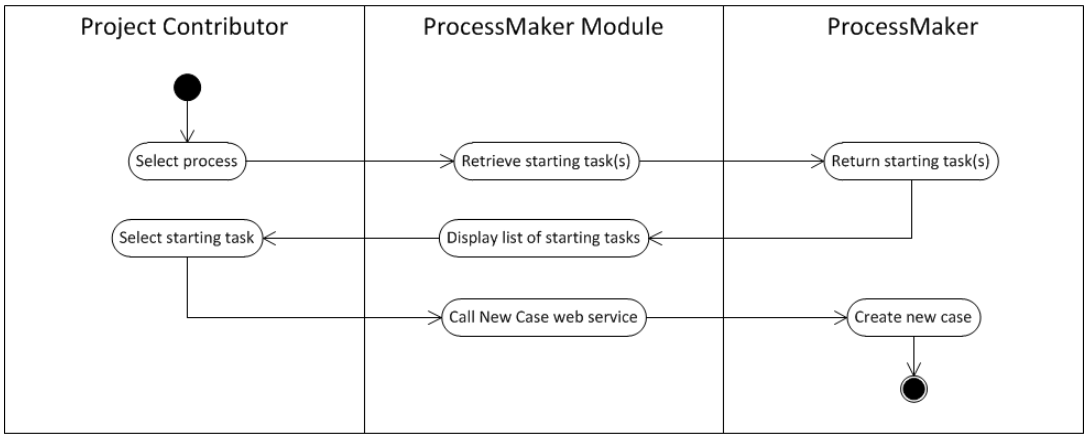


Figure 18: Initiate New Case activity diagram

B..1.4 Respond to Case A project contributor can respond to a case by completing the task assigned to him or modifying his draft cases. Figure 19 shows the Respond to Case use case diagram.

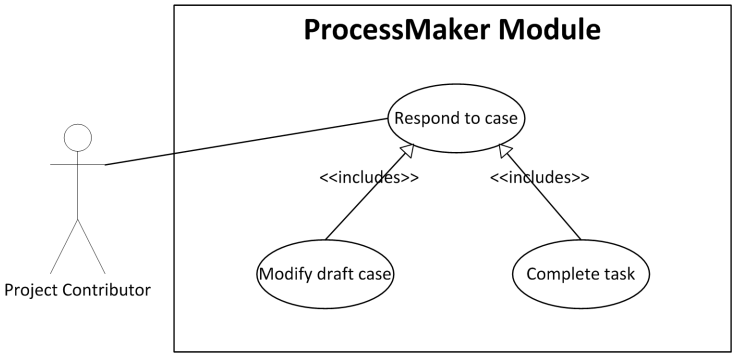


Figure 19: Respond to Case use case diagram

B..1.4.1 Modify Draft Case An opened case with an unfinished task sends the case into the Drafts category. The project contributor can modify a draft case to complete the task and pass on the case. Figure 20 shows the Modify Draft Case activity diagram.

B..1.4.2 Complete Task The project contributor can complete the task assigned to him in a case. Figure 21 shows the Complete Task activity diagram.

B..1.5 View Case A project contributor can view the information and documents of cases that he has been a part of. He may also view cases by category.

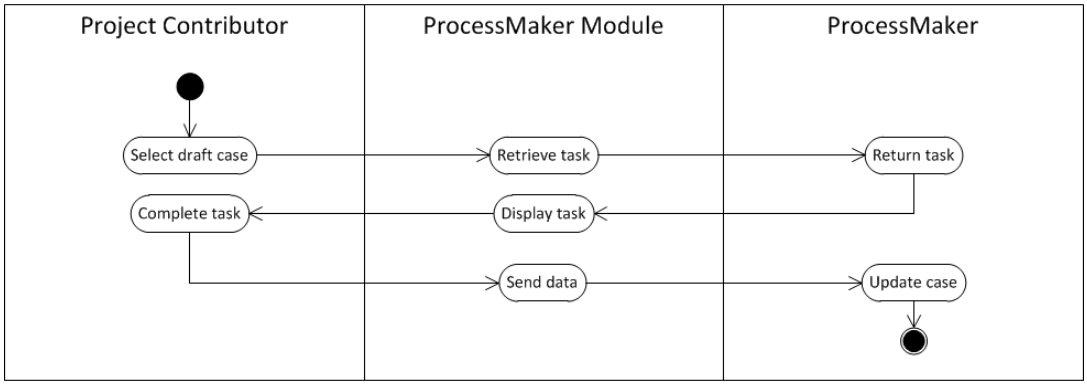


Figure 20: Modify Draft Case activity diagram

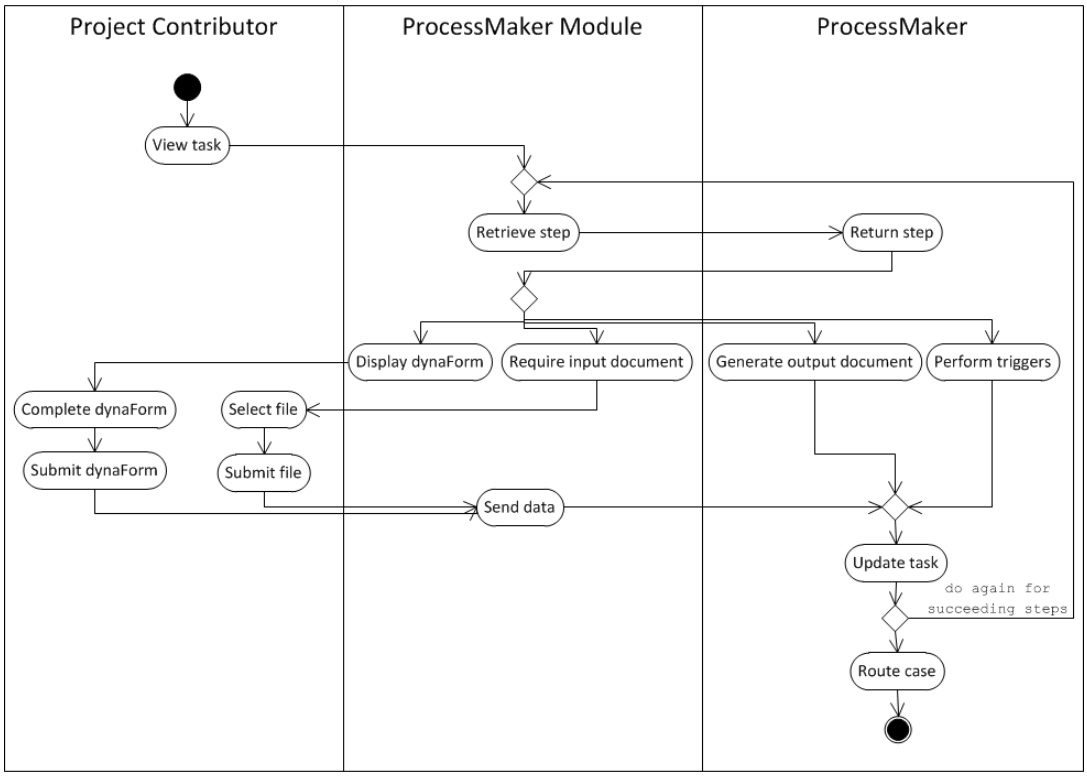


Figure 21: Complete Task activity diagram

Figure 22 shows the View Case use case diagram.

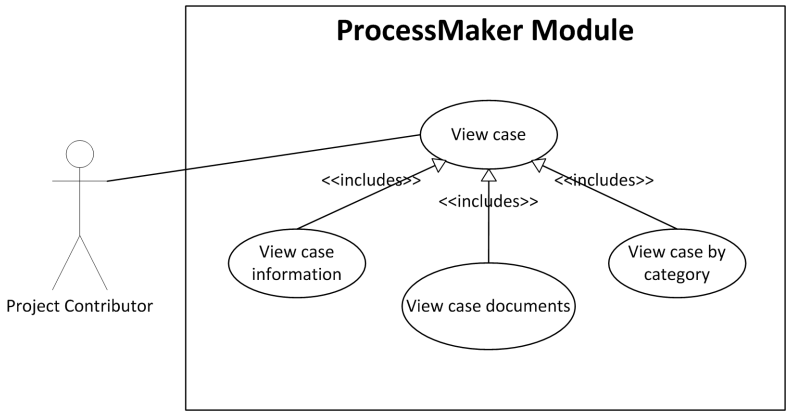


Figure 22: View Case use case diagram

B..1.5.1 View Case Information The project contributor can view the details of a particular case such as its process map and information. Figure 23 shows the View Case Information activity diagram.

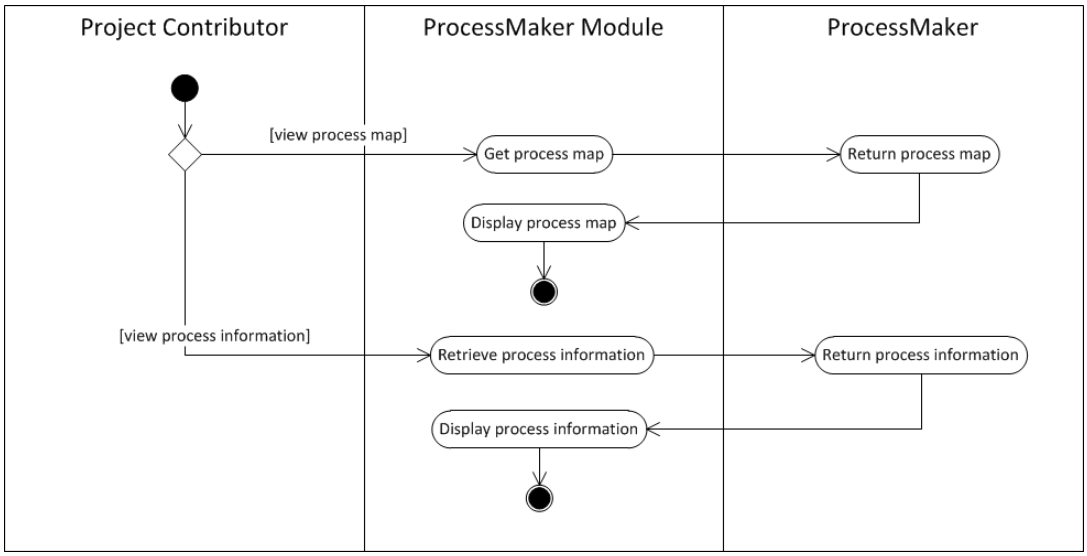


Figure 23: View Case Information activity diagram

B..1.5.2 View Case Documents The project contributor can view uploaded and generated documents for a particular case. Figure 24 shows the View Case Documents activity diagram.

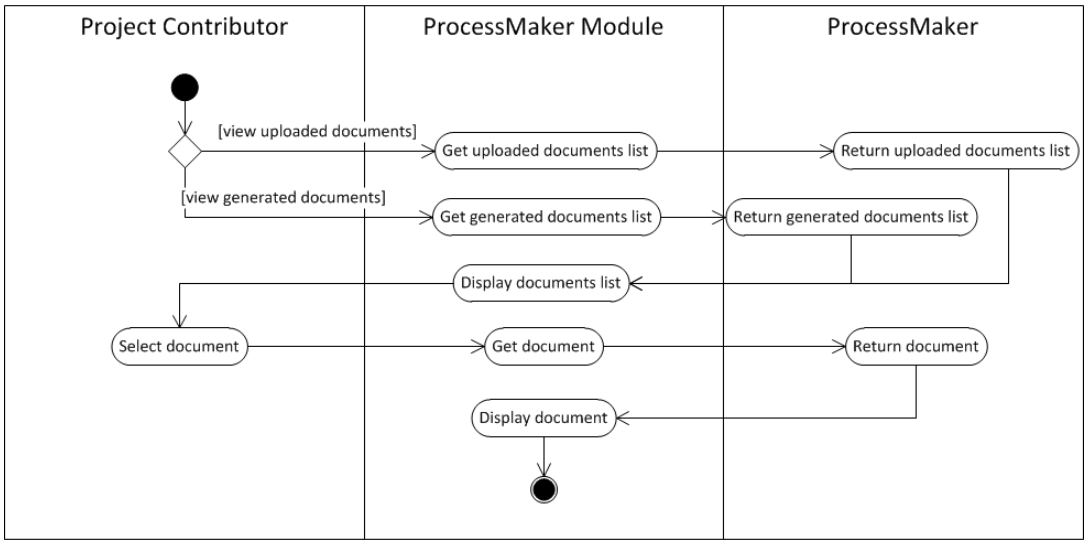


Figure 24: View Case Documents activity diagram

B..1.5.3 View case by Category The project contributor can view his cases by category – Inbox, Draft, Participated, Unassigned, or Paused. Figure 25 shows the View Case by Category activity diagram.

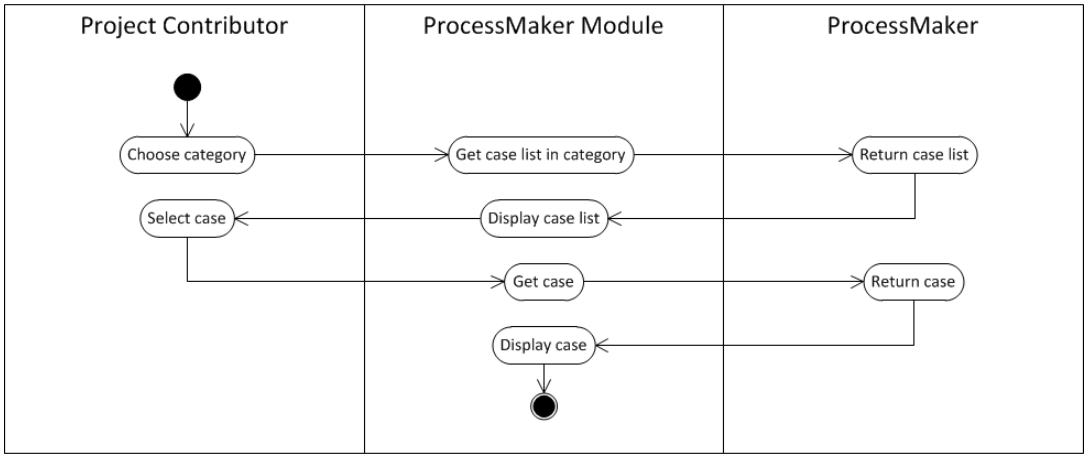


Figure 25: View Case by Category activity diagram

B..2 Access Project

The project contributor has the privelege of managing miscellaneous files, tools, and references. He can also participate in forums as well as access files in the Subversion repository and the new module, Git repository. The restricted user on the other hand will only be able to download the materials in the project. Figure 26 shows the Access Project use case diagram with the newly added functionality

– Access files in Git repository.

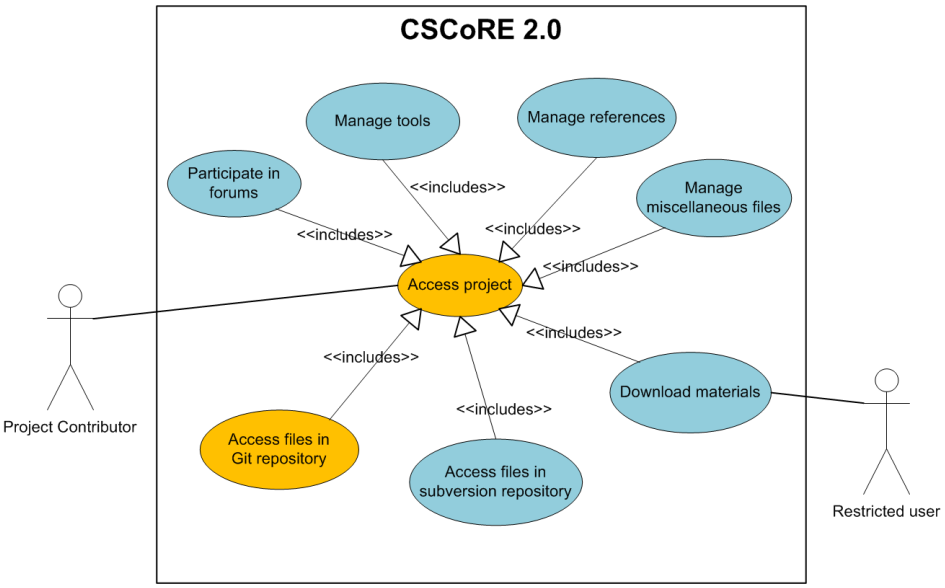


Figure 26: Access Project use case diagram

B..2.1 Access Files in Git Repository A project contributor will be able to browse files of a branch, clone the repository, view changes done in CSCoRE, pull from a remote repository and push changes into the repository. Figure 27 shows the Access files in Git repository use case diagram, Figure 28 shows the Browse Files and Branches activity diagram, Figure 29 shows the Clone Repository activity diagram, Figure 30 shows the View Change Log activity diagram, Figure 31 shows the Pull from Remote Repository activity diagram, and finally Figure 32 shows the Push Changes activity diagram.

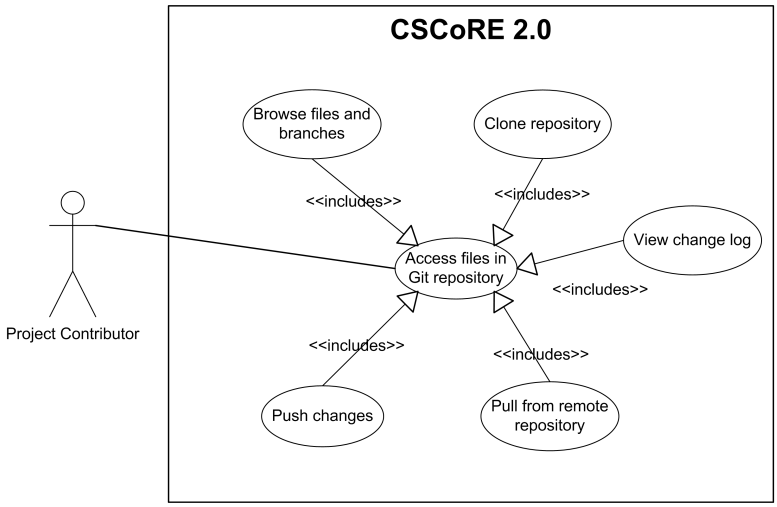


Figure 27: Access files in Git repository use case diagram

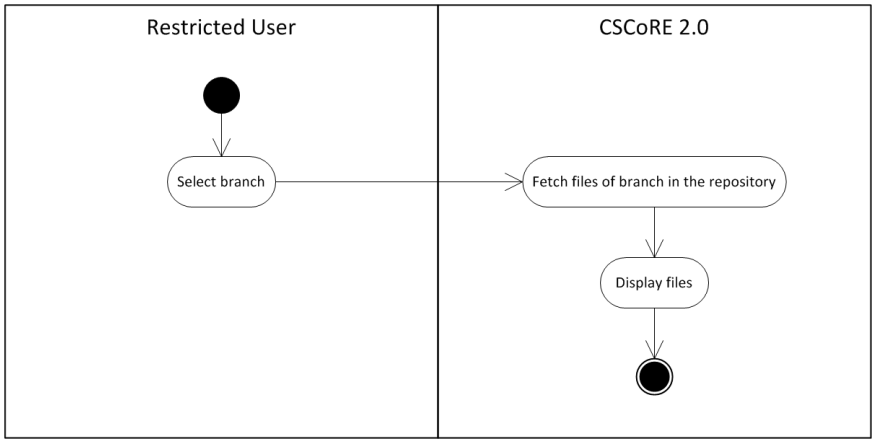


Figure 28: Browse Files and Branches use case diagram

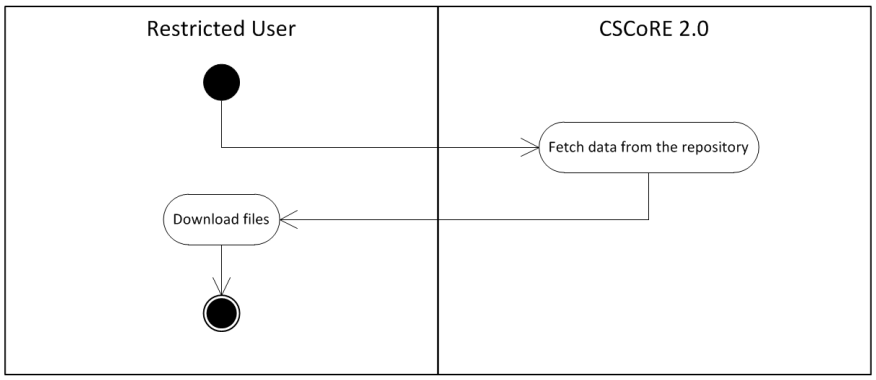


Figure 29: Clone Repository use case diagram

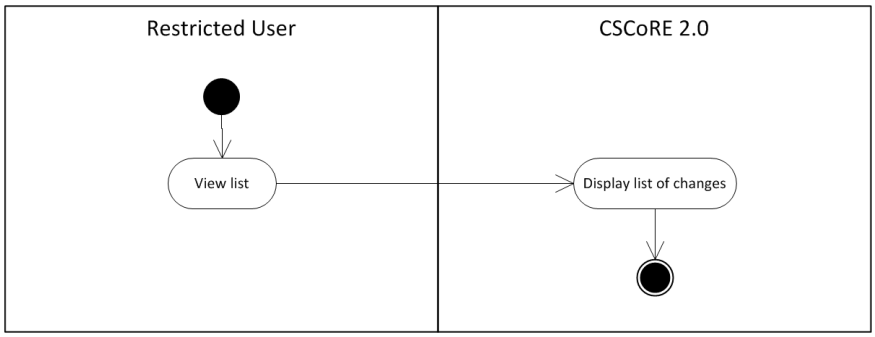


Figure 30: View Change Log use case diagram

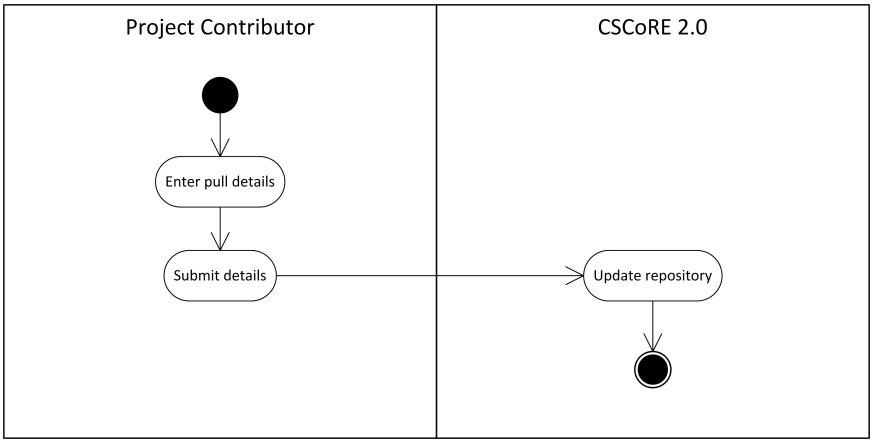


Figure 31: Pull from Remote Repository use case diagram

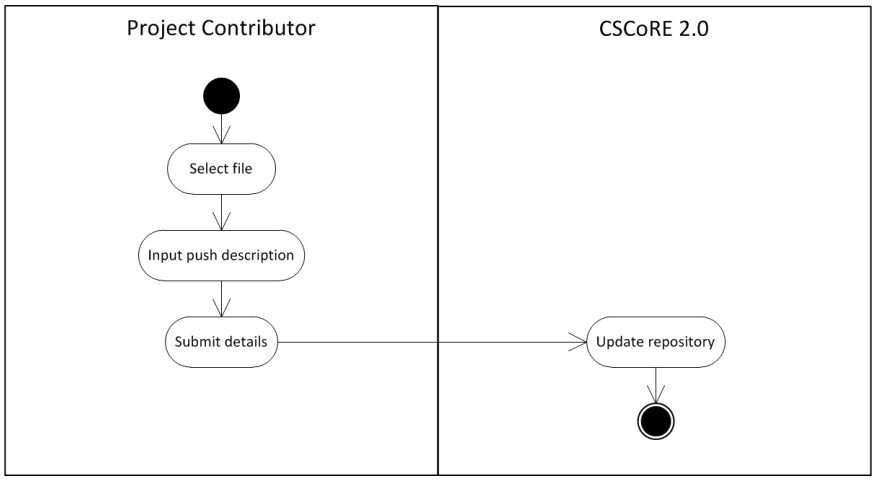


Figure 32: Push Changes use case diagram

V. Architecture

A. System Architecture

CSCoRE is built using the Google Web Toolkit framework. Its presentation layer is comprised of GWT user interface components namely HTML, CSS and JavaScript. The JavaScript is obtained from the compilation of Java codes by GWT. It then connects with Java servelets that utilize the application programming interfaces (API) for communicating with the storage layer. MySQL Query API is used to communicate with the system database and SVNKit is the API used to connect to the subversion repository. For CSCoRE 2.0 specifically, ProcessMaker Web Services API is employed to communicate with the ProcessMaker application while JGit API is utilized to connect to the Git repository. Figure 33 shows the architecture of the system.

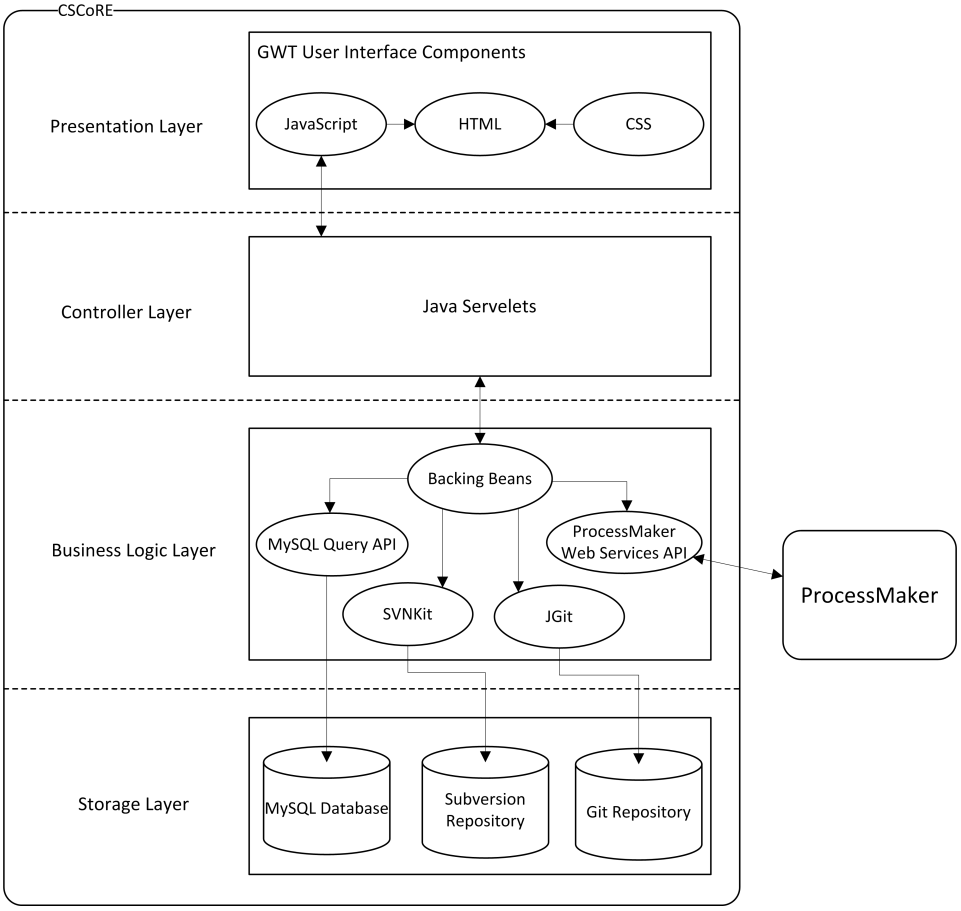


Figure 33: CSCoRE system architecture

A..1 ProcessMaker Web Services

ProcessMaker Web Services provide an interface to effectively integrate ProcessMaker with external systems. Compliant with Simple Object Access Protocol (SOAP) 1.2 and Web Services Description Language (WSDL) 1.1 as defined by the World Wide Web Consortium (W3C), it allows for functions and objects to be generated in languages like Java which has SOAP and WSDL libraries. ProcessMaker features such as the management of users, groups, processes and cases can thus be remotely accessed.

A..2 JGit

JGit is a lightweight pure Java library implementing the Git version control system. It provides methods for repository access routines, network protocols and core version control algorithms to perform commands on Git repositories. The initial purpose of this API was for the development of a plugin for the Eclipse integrated development environment (IDE) to work with Git source code management. Its adoption has since expanded to other applications that host and manage source code maintained in Git repositories [41].

B. Technical Architecture

B..1 Server

B..1.1 Hardware Requirements

- 1.6 GHz processor
- 512MB RAM
- 100MB free disk space

B..1.2 Software Requirements

- OS: Ubuntu 12.04.2
- Apache Tomcat 6.0
- MySQL 5.0
- Java Runtime Environment 1.6
- ProcessMaker 2.0.45 (already installed and set up)

B..2 Client

CSCoRE requires the client to have a web browser. It works on:

- Google Chrome 26.0.1410.43
- Mozilla Firefox 11.0

VI. Results

A. ProcessMaker Module

The ProcessMaker module loads the workflow application within the page of CSCoRE. Upon initial use of the ProcessMaker module, database information is needed to setup the workspace. If the user is a project administrator, the settings form can be filled up with the details of three databases as shown in Figure 34.

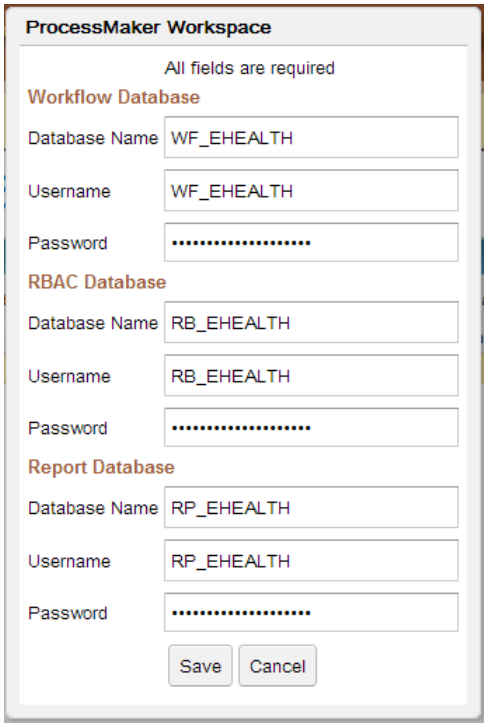


Figure 34: ProcessMaker Workspace Settings

The project administrator can manage the users and groups in ProcessMaker. Figure 35 displays the list of users. These users can be added to groups as illustrated by Figure 36.

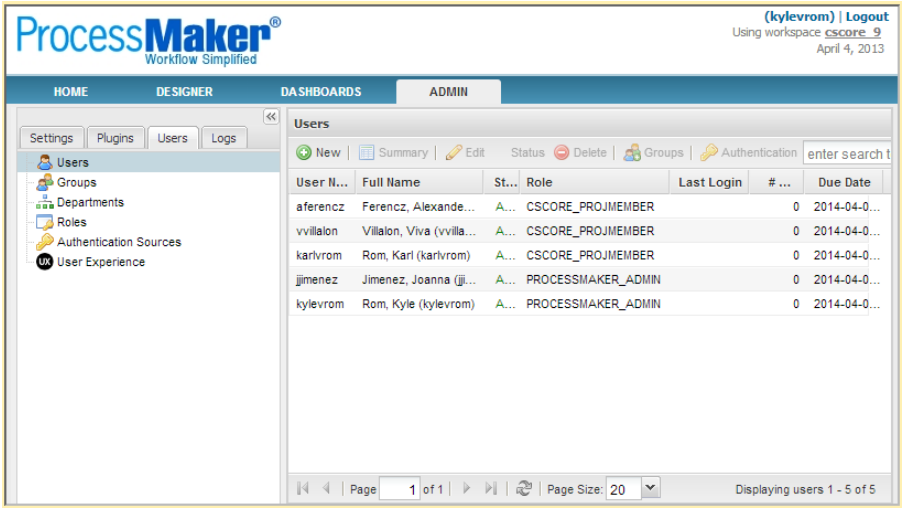


Figure 35: ProcessMaker users page

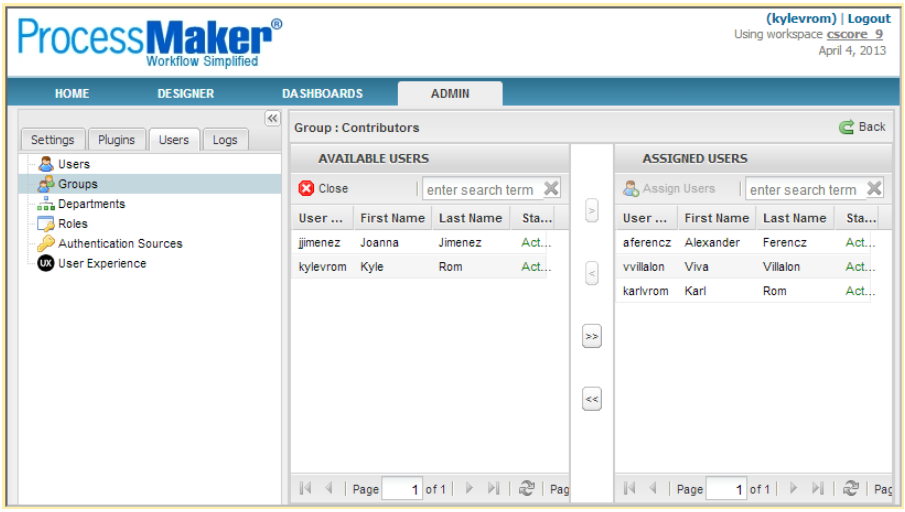


Figure 36: Adding users to groups

Administrators can also import a process as shown in Figure 37. Conversely, existing processes may be deleted like in Figure 38.

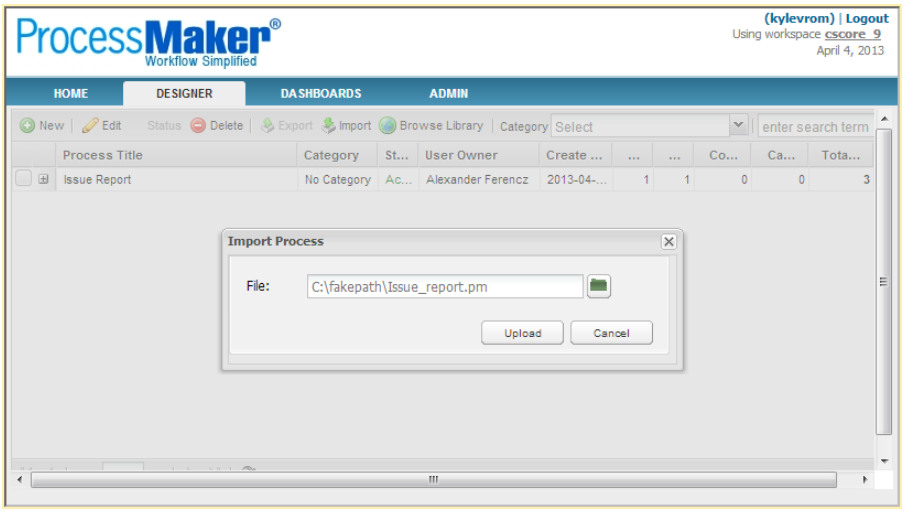


Figure 37: Import a process

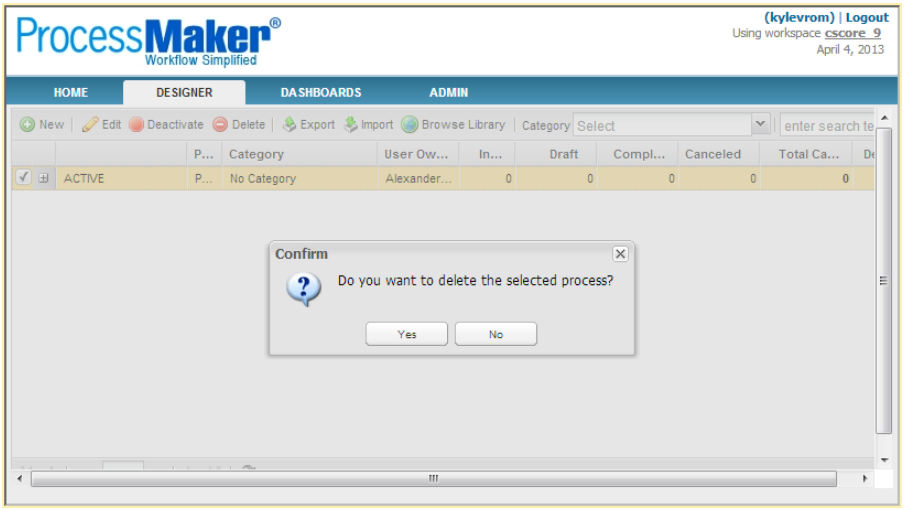


Figure 38: Delete a process

Contributors of this module can initiate a case in which they have been assigned in the first task as in Figure 39. Likewise, they can respond to a case routed to them and complete the steps in that particular task. Figure 40 illustrates a Dynaform which is one type of step in ProcessMaker.

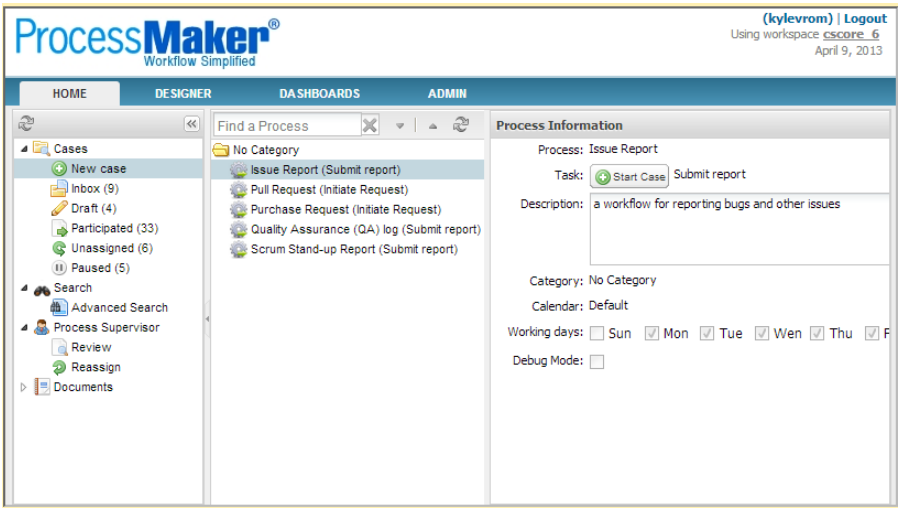


Figure 39: Initiate a new case

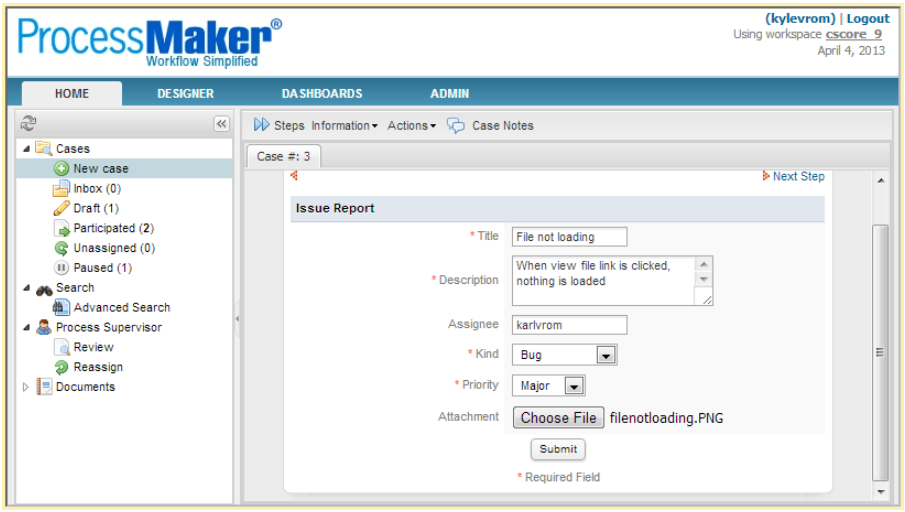


Figure 40: Respond to a case

All the authorized members – administrators and contributors – can view their inbox (Figure 41), draft (Figure 42), participated (Figure 43), unassigned (Figure 44) and paused (Figure 45) cases as well as the related documents (Figure 46) through the links found on the side panel under the Home tab.

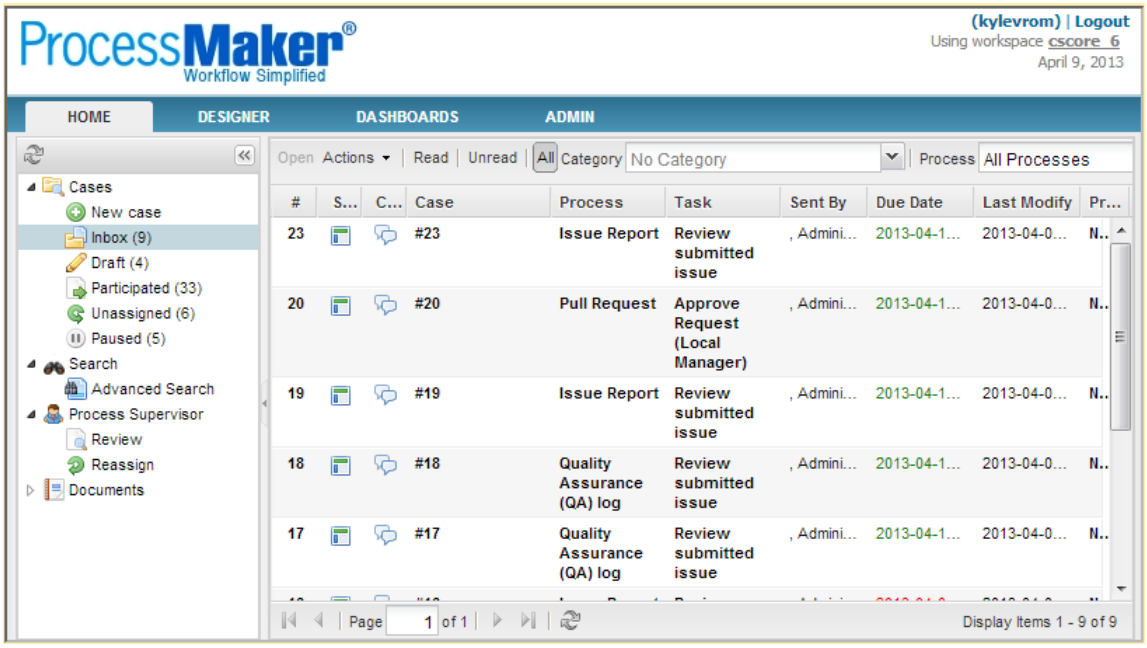


Figure 41: View cases in inbox

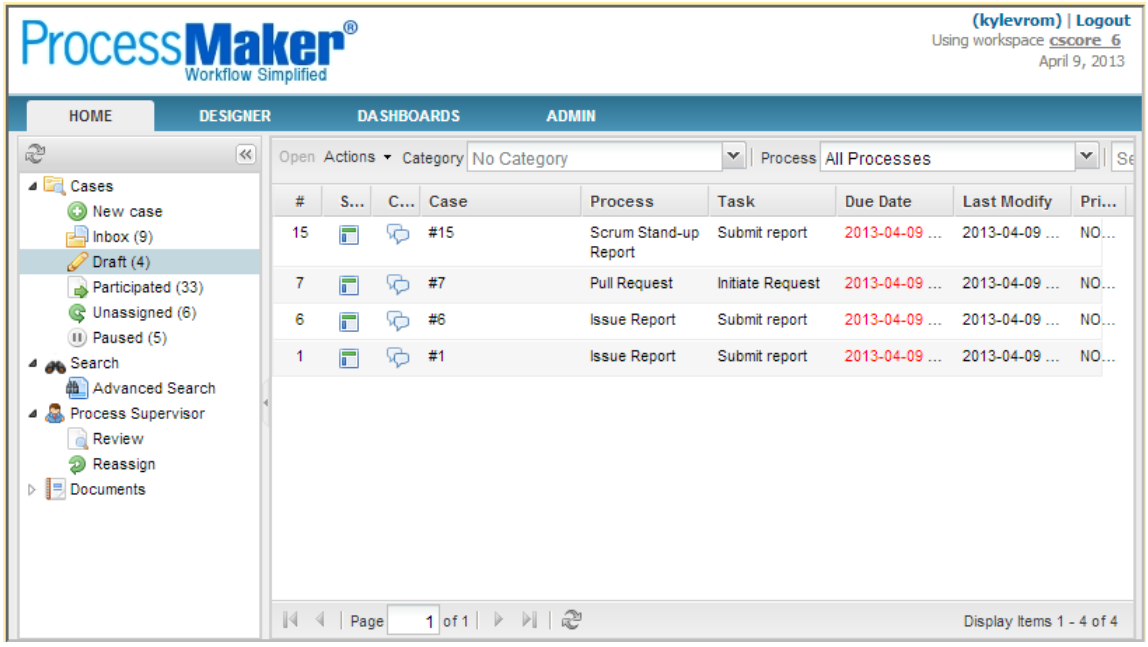


Figure 42: View draft cases

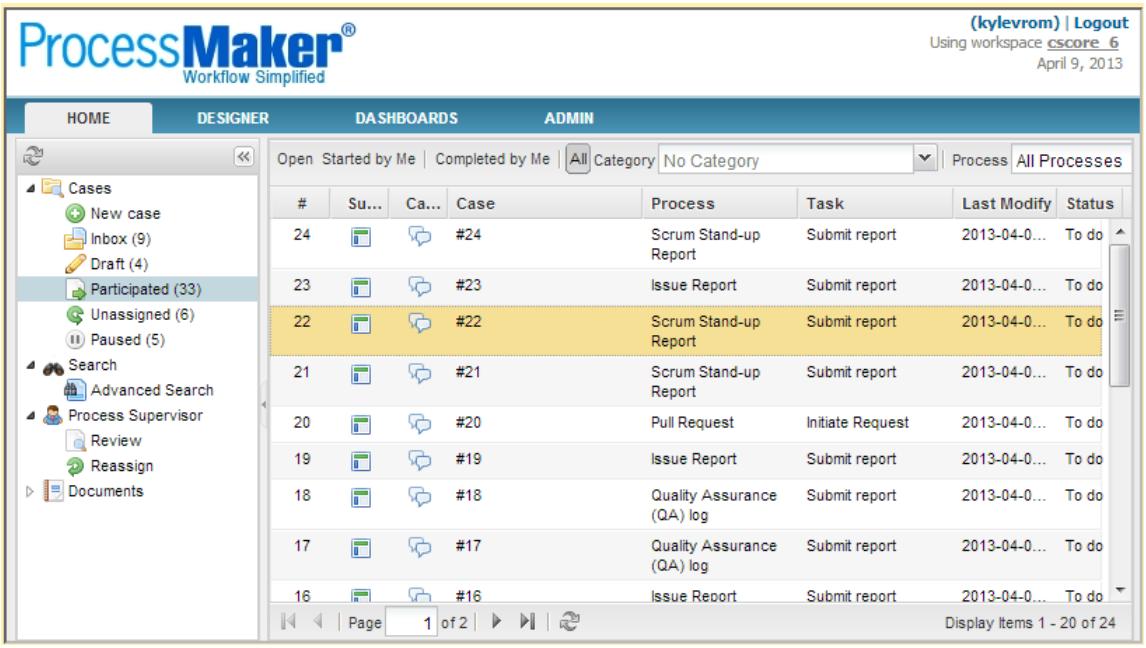


Figure 43: View participated cases

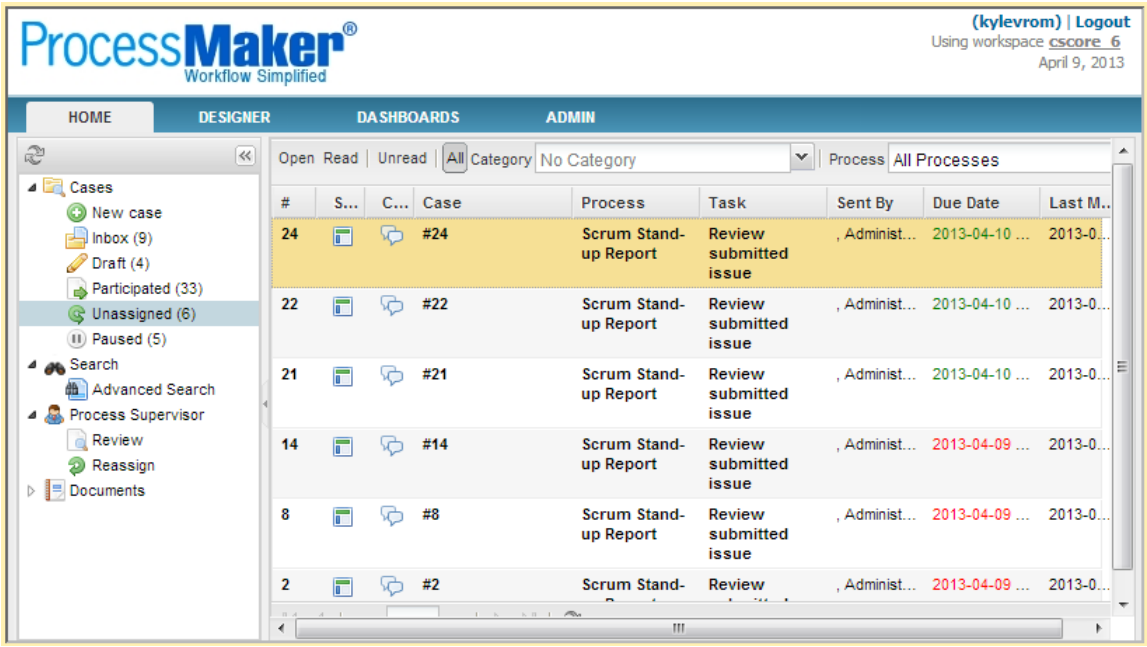


Figure 44: View unassigned cases

Figure 45: View paused cases

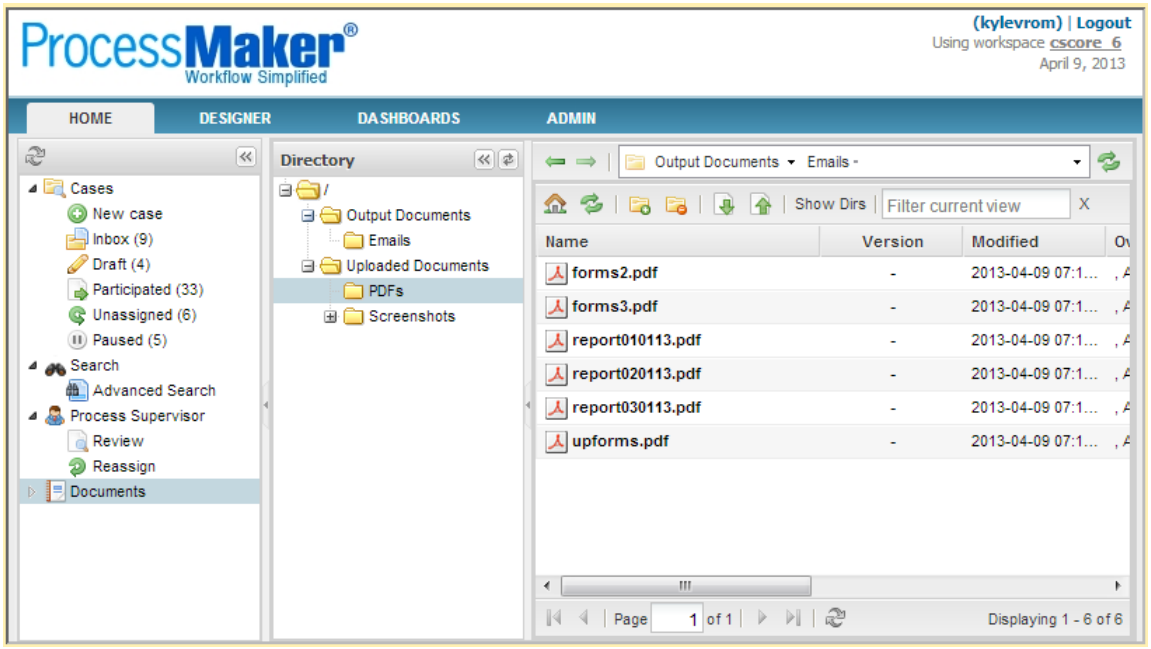


Figure 46: View documents

B. Git Repository Module

The Git repository module gives access to the Git repository for the project. The project administrator can give the settings required to connect to the remote repository. Figure 47 shows the settings dialog box.

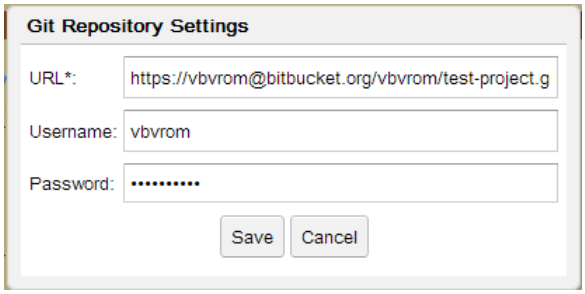


Figure 47: Git repository settings

All members can browse the repository directory like in Figure 48. The user can switch branches and view the files and folders contained. Clicking on a filename permits the user to view the commit details and contents of the file. Figure 49 displays the file page.

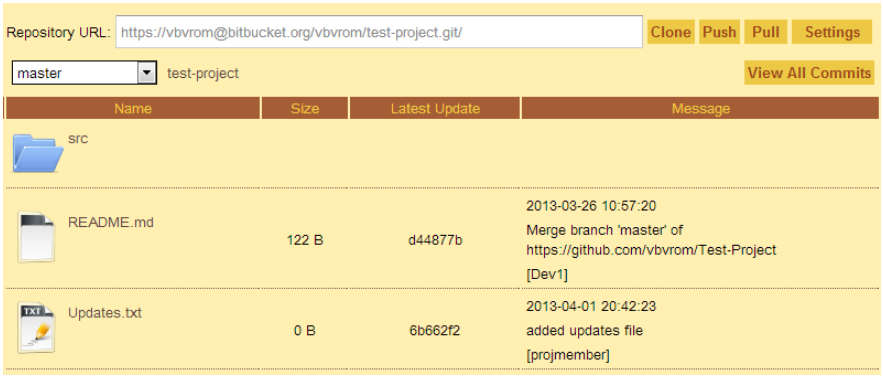


Figure 48: Browse Git repository

The list of all commits done in the repository can also be viewed. Figure 50 depicts the commit history.

All users can also clone the repository. The Save As dialog box as shown in Figure 51 opens in order for the user to download the zipped file of the repository.

Moreover, for contributors, the Push and Pull options are also available. Figure 52 gives the Push details dialog box and Figure 53 displays the Pull details dialog box.

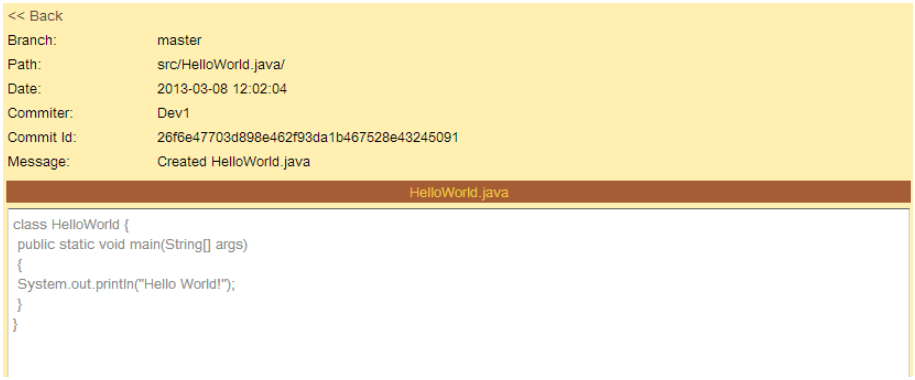


Figure 49: View a file

Date	Author	Commit	Message	
2013-04-01 20:42:23	projmember	6b662f2	added updates file	Browse repository
2013-03-26 11:25:16	Dev1	00e0cae	initial commit for core files Calc.java and Utils.java	Browse repository
2013-03-26 10:57:20	Dev1	d44877b	Merge branch 'master' of https://github.com/vbvrom/Test-Project Conflicts: README.md	Browse repository
2013-03-26 10:49:39	vbvrom	28263bb	Initial commit	Browse repository
2013-03-22 10:20:43	Dev1	1ada815	created testing file.txt	Browse repository
2013-03-17 19:36:26	Dev1	864136a	Added username input in HelloWorld.java Added Person.java	Browse repository
2013-03-08 12:12:32	Dev1	c173a5c	Added UserManual.txt	Browse repository
2013-03-08 12:02:04	Dev1	26f6e47	Created HelloWorld.java	Browse repository
2013-03-07 02:17:06	Dev1	6ab6386	First commit. Adding a README.	Browse repository

Figure 50: Git commit history

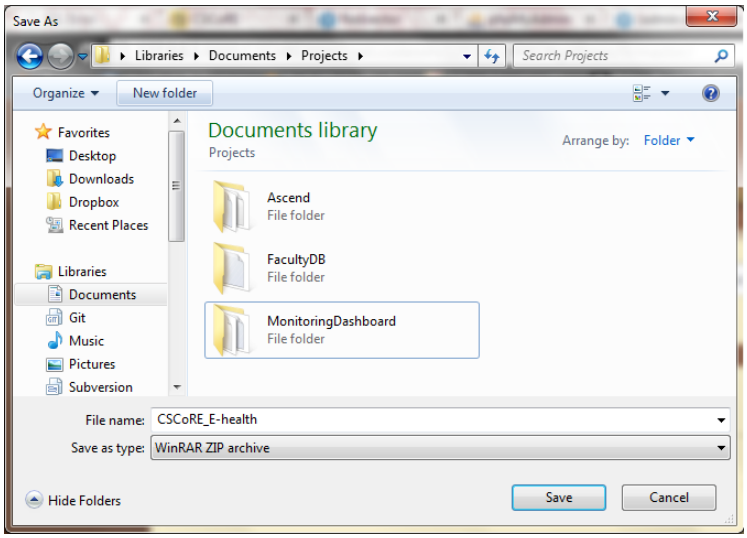


Figure 51: Downloading the Clone

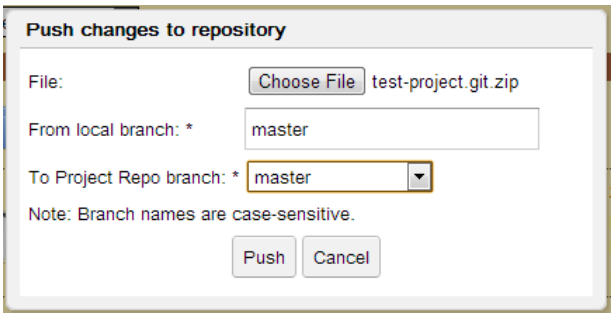


Figure 52: Push changes to repository

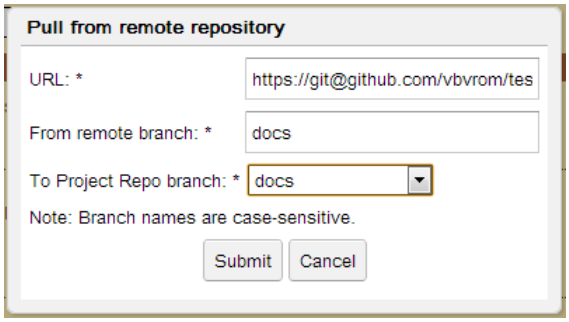


Figure 53: Pull updates from remote repository

VII. Discussions

A. ProcessMaker Module

The ProcessMaker module displays the interface of the said application embedded within CSCoRE so as to fully take advantage of the functionalities that the workflow management system offers. Project contributors and administrators are able to interact within this space for running workflow cases requiring their participation.

Each project in CSCoRE is accommodated by creating its own workspace on ProcessMaker. A workspace provides an environment for the processes and users of a project to be contained seperately from all those of other projects. This eliminates the possible confusion that may arise if all projects share one workspace where the resources like documents and databases are collectively shared. It also helps the administrators oversee the activities within the project.

ProcessMaker by default creates three databases for the operation of a workspace. However, this becomes problematic for the server administrator since this requires that the application be given root access for the setup of these databases without the administrator having control over it. To address this issue, PHP scripts have been written and packaged into a plugin to incorporate them into the workflow management application and bypass the usual procedure. With this solution, the setup of a workspace asks for three unique and empty databases – Workflow, Role-based Access Control and Reports (with the preferred naming conventions WF_[ProjectName], RB_[ProjectName] and RP_[ProjectName], respectively) to be provided by the server administrator.

In addition, the plugin also contains code for the creation of a new user role with permissions picked to comply with CSCoRE. This enables ProcessMaker to differentiate a contributor from an administrator. This and the use of web services allow smooth connection between the two systems in the management of users.

The said plugin was specifically built for CSCoRE using ProcessMaker's Gul-

liver script for plugin development [42]. This allows the necessary source code addition used by CSCoRE to be applied to subsequent versions of Processmaker. Upon initial installation of CSCoRE, the CSCoRE plugin must also be imported by the system administrator by logging in ProcessMaker as an administrator and going to the Plugins Manager under the Admin tab as shown in Figure 54. The installed plugin contains code for generating a new role named CSCORE_PROJMEMBER in addition to three important PHP files that are copied into the ProcessMaker installation directory: cscore_newSiteProxy.php, class.cscore_Installer.php and mysqldump.php. Cscore_newSiteProxy.php is called by CSCoRE on a POST request for creating a workspace and fetching or updating database details. Class.cscore_ Installer.php is a variation of the default script for creating a workspace which works by setting up a workspace using the provided databases instead of generating new ones. Mysqldump.php facilitates the configuring of databases.

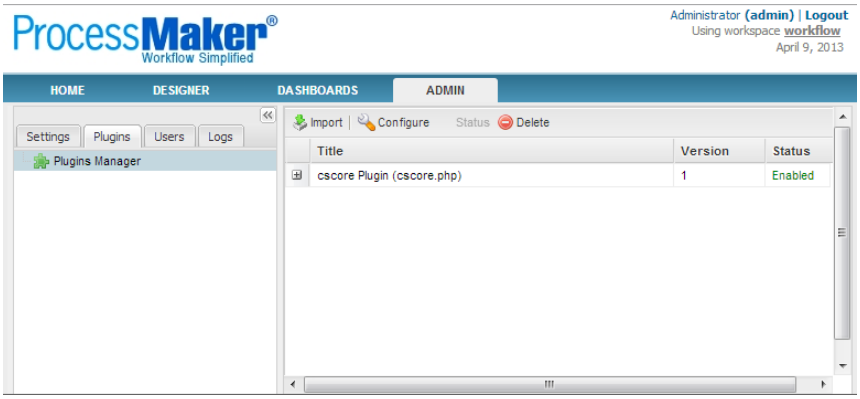


Figure 54: ProcessMaker Plugins Manager

With regard to processes, ProcessMaker process definition files previously made can be imported. Assignment of members to user groups for the process tasks can be done by the administrator within the graphical user interface of the module. After this, process cases can now be run. Although for some web servers hosting ProcessMaker the Designer tab fails to load properly, this does not hinder the usage of imported processes provided that their process tasks are assigned to groups to which users can be added.

B. Git Repository Module

The Git repository module connects to a web-based remote repository and displays its contents for browsing. It supports the three essential commands for performing collaboration on a repository: Clone, Push and Pull. It also permits viewing of the commit history of the repository.

The repository browser allows the user to switch between branches to view their files and directories. Clicking a file will show the user details on its commit as well as the contents of the file. From the commit history, all commits are listed and the repository can be browsed at a specific point in time.

The Clone command is performed by letting the user download a zipped copy of the whole repository. Working with a zipped file was necessary because of the security constraints of web browsers which block an application from directly accessing the file system of a local computer.

The Push command works in a similar manner by asking for a zipped file of the repository containing new commits. The supported functionality of Push is that changes from one branch of the uploaded repository copy are then pushed or joined to another branch in the project repository in CSCoRE. Upon uploading the zipped file, CSCoRE first verifies that all commits to be pushed are done by the user by checking the committer identity of each commit before going through with the push. This is made possible by the feature of Git on doing commits locally before they are merged with a remote repository. Comparing this with SVN where commits are done directly at the central repository, this feature turns out to be an advantage of Git since it provides an inherent way for the verification of commits without the need for additional access control mechanisms.

Moreover, the Pull command for this module is done by fetching and merging updates from a branch of another remote repository specified by a URL into a branch in the CSCoRE repository.

Finally, the list of all commits offers a way to track the changes that have been done in the repository. Through this, the user is able to view the state of the

repository at a chosen commit. The challenge here is in mapping the name of the commit authors to the names of members of the project. It is currently handled by making sure that users upload commits that are named after them.

VIII. Conclusions

The Computer Science Collaborative Research Environment (CSCoRE) 2.0 gains a step up on its purpose of being a beneficial tool for collaboration specifically for projects in the field of Computer Science through the addition of the modules ProcessMaker and Git repository.

ProcessMaker provides CSCoRE with an embedded workflow management system in which pertinent workflows tailored to the needs of a project can be utilized to streamline the flow of information among members of the team. This is made possible through workflow automation that routes tasks to the users required to perform them. By following the specifications contained in the imported process file, procedures are thus done systematically with all needed information gathered through forms and documents while delays between tasks consequently reduced. The Git repository on the other hand, offers users the ability to conveniently browse and obtain version controlled files of a project. More importantly, the distributed nature of Git gives enough flexibility to fit well with the structure on how most software development teams collaborate and contribute. Through this, contributors are able to maintain and update the hosted source codes while other interested members can obtain a copy of the files to try out for themselves.

IX. Recommendations

These novel modules incorporated in CSCoRE 2.0 can be both improved further in the aspects of functionality and ease of use. For instance, pre-made comprehensive processes defined in .pm files can be bundled with the system to serve as a template for common workflows like issue tracking or as a quick start to using the module. ProcessMaker can also be made to integrate deeper than the current implementation by creating an interface that allows the other modules of CSCoRE to interact in processes while still harnessing ProcessMaker's workflow engine through web services. Similar workflow managements systems like YAWL can also be added to CSCoRE. The Git repository meanwhile can be extended by supporting the various options of Push and Pull commands. Mechanisms for access control or authorization of users who update is also highly suggested. Syntax highlighting of contents can also help to make files more readable. Finally, other Git commands may also be added such as Branch to facilitate separate development of features, Tag to mark stable versions, Diff to compare versions of files, and Blame to see who committed a specific change in a file.

X. Bibliography

[1] J. S. Katz and B. R. Martin, "What is research collaboration?," *Research Policy*, vol. 26, pp. 1–18, 1997.

[2] "Benefits of research collaboration." `http://www.rcr.emich.edu/module9/i4\_benefits.html/`, December 2011.

[3] W. M. Coalition, "Terminology and glossary," tech. rep., WfMC, 1999.

[4] "The key benefits of workflow." `http://www.e-workflow.org/`, December 2011.

[5] P. Santos, "Distributed version control systems in the enterprise." `http://www.infoq.com/articles/DVCS-Enterprise`, May 2012.

[6] I. Skerrett, "Community survey result." `http://ianskerrett.wordpress.com/2012/06/08/eclipse-community-survey-result-for-2012/`, June 2012.

[7] I. Clatworthy, "Distributed version control systems – why and how," tech. rep., Open Source Developers Conference, 2007.

[8] J. Jimenez, "Cscore: Computer science collaborative research environment," tech. rep., University of the Philippines - Manila, 2012.

[9] K. R. Abbott and S. K. Sarin, "Experiences with workflow management: issues for the next generation," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, CSCW '94, (New York, NY, USA), pp. 113–120, ACM, 1994.

[10] H. A. Reijers, M. Song, and B. Jeong, "Analysis of a collaborative workflow process with distributed actors," *Information Systems Frontiers*, vol. 11, pp. 307–322, July 2009.

[11] D. Georgakopoulos, M. Hornick, and A. Sheth, "An overview of workflow management: from process modeling to workflow automation infrastructure," *Distrib. Parallel Databases*, vol. 3, pp. 119–153, April 1995.

[12] X. Ying, C. Deren, and C. Min, "Research of web services workflow and its key technology based on xpdl," *2004 IEEE International Conference on Systems, Man and Cybernetics*, 2004.

[13] D. K. C. Chan and K. R. P. H. Leung, "Software development as a workflow process," in *Proceedings of the Fourth Asia-Pacific Software Engineering and International Computer Science Conference*, APSEC '97, (Washington, DC, USA), pp. 282–, IEEE Computer Society, 1997.

[14] C. Mao and J. Zhang, "Workflow-based testing process management of software project," *WiCom 2007 International Conference on Wireless Communications, Networking and Mobile Computing*, 2007.

[15] Y. Chen and J. Xin, "A process-oriented configurable workflow system model for cooperative project management," *2010 14th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2010.

[16] A. Sheth, "From contemporary workflow process automation to adaptive and dynamic work activity coordination and collaboration," *SIGGROUP Bull.*, vol. 18, pp. 17–20, December 1997.

[17] A. Haller, M. Marmolowski, W. Gaaloul, E. Oren, B. Sapkota, and M. Hauswirth, "From workflow models to executable web service interfaces," in *Proceedings of the 2009 IEEE International Conference on Web Services*, ICWS '09, (Washington, DC, USA), pp. 131–140, IEEE Computer Society, 2009.

[18] C. Pappas, E. Coscia, G. Dodero, V. Gianuzzi, and M. Earney, "A mobile e-health system based on workflow automation tools," in *Proceedings of the 15th*

IEEE Symposium on Computer-Based Medical Systems (CBMS'02), CBMS '02, (Washington, DC, USA), pp. 271–, IEEE Computer Society, 2002.

[19] D. Miers, "Sharepoint as a strategic weapon: Beyond content management and simple workflow," tech. rep., BPM Focus, 2009.

[20] U. Software, "Business process acceleration within oracle e-business suite," tech. rep., UC4 Software, 2010.

[21] B. Silver, "Sap netweaver bpm," tech. rep., Bruce Silver Associates, 2009.

[22] B. Wire, "Processmaker bpm is one of first applications to be certified through the intel certified solutions program," tech. rep., Business Wire, 2008.

[23] Processmaker, "Case study: De una," tech. rep., Processmaker, 2011.

[24] Processmaker, "Case study: Nuevatel sa," tech. rep., Processmaker, 2011.

[25] Processmaker, "Case study: Edautomate," tech. rep., Processmaker, 2011.

[26] S. Azad, "Svn vs. git: Who will be the future of revision control?." `http://www.richappsconsulting.com/blog/blog-detail/svn-vs-git-who-will-be-the-future-of-revision-control/`, December 2008.

[27] "Compare repositories." `http://www.ohloh.net/repositories/compare`, October 2012.

[28] E. Newren, "Gnome dvcs survey results." `http://blogs.gnome.org/newren/2009/01/03/gnome-dvcs-survey-results/`, January 2009.

[29] G. Oteniya and A. Ijo, "Workflow and business process management for eiectronc government," tech. rep., UNU-IIST, Macao, 2011.

[30] R. Software, "Rational unified process: Best practices for software development teams," tech. rep., Rational Software, 1998.

[31] "Generic software development workflow." http://pnexpert.com/Typical_Development_Workflow.html, February 2012.

[32] H. Delugach, "Formal analysis of workflows in software development," *Handbook of Research on Socio-Technical Design and Social Networking Systems*, 2009.

[33] "Open source workflow software & business process management bpm." http://www.processmaker.com, February 2012.

[34] "Processmaker architecture." http://wiki.processmaker.com/index.php/ProcessMaker_Architecture_Diagrams, February 2012.

[35] "Managing users." http://wiki.processmaker.com/index.php/2.0/Managing_Users, February 2012.

[36] "Process object model." http://wiki.processmaker.com/index.php/Object_Model_for_Process_definition, February 2012.

[37] D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard, "Web services architecture," tech. rep., W3C, 2004.

[38] E. Cavanaugh, "Web services: Benefits, challenges, and a unique, visual development solution," tech. rep., Altova, 2006.

[39] "Introduction to git." http://learn.github.com/p/intro.html, 2010.

[40] S. Chacon, *Pro Git*. Berkely, CA, USA: Apress, 1st ed., 2009.

[41] "Jgit - java git." http://www.jgit.org/, 2012.

[42] "Plugin development." http://wiki.processmaker.com/index.php/2.0.36/Plugin_Development, November 2012.

XI. Appendix

A. Source Code

XII. Acknowledgement

Finally! My time for making this SP has stretched fairly long enough I had nightmares on not seeing this through. Nevertheless, a huge sigh of relief for it is now done. Although this final requirement for my degree was to be made individually, I feel that I have never been alone in the struggle. So I would like to express my utmost gratitude, in efforts that I hope would have bearing, to the many people who have been with me as I strutted (and almost crawled) my way to the finish line.

First of all, a very loud thank you with matching tears of joy to the most patient SP adviser ever, Sir Richard Bryann Chua! It is really commendable how he is able to generously give precious time and eagerly dispense the needed push to help his advisees succeed in this endeavor. I may have consistently tested his patience and tolerance each consultation but a million thanks Sir for not giving up on me. I feel so fortunate and I do sincerely appreciate it. Doing this SP under his guidance has thought me a great deal, not only within but also well beyond the field of Computer Science.

A big warm hug is for Joanna Marielle Jimenez, author of the first version of CSCoRE and Muhmi JJ of ComSci Dose '08. She has always been available to help as I develop this version two of the system and was even there to give tips on anything related to the SP, from start until the end. Like a true mommy of the block, she showered each one of us with care and support and for that we love her very much.

My heartfelt gratitude also goes out to my unfalteringly supportive family. I may have been away from home, far from where my parents and siblings are, yet I know that they will always be cheering on me. I am truly grateful for Mamae and Papa who have worked their hardest to fund my studies and who have been very understanding during the time of making my SP. Of course to my grandma, Mama Mary, for taking care of me and praying most ardently that I accomplish this in due time thank you oh so much.

My gratefulness also extends to the University, my professors and to my blockmates, ComSci Dose '08. Thanks to all these good friends who have stuck through the journey to laugh alongside and to learn all the while.

Above all, it is with the power of our Almighty God that the highest praises should be made. Thank You very much for granting the wisdom, strength of will, loving inspiration, and the presence of these people throughout these years, most especially for the accomplishment of this special problem. For hearing my prayers and showering timely miracles my way, I would like to humbly raise this work for Your glory.