A Conceptual Framework for Collaboration Systems

Yeh, Chao-Chin[1] **Chen, Huei-Huang**[2]

[1]Yeh, Chao-Chin (jao_chin@so-net.net.tw)
[2]Chen, Huei-Huang (hhchen@ttu.edu.tw)
[1,2]Department of Computer Science and Engineering, Tatung University

Abstract

This paper proposes a conceptual framework for collaboration system as reference architecture for the collaboration systems and illustrates the functionalities of a collaboration system and the issues should be considered in developing such a collaboration system.

The conceptual framework for collaboration systems proposed in this paper includes 5 logical layers (the IMPCA framework) through clustering the functional modules to layers. This paper illustrates the responsibilities and functionalities of every module and layer and proposes a possible solution for the IMPCA framework which integrates many open source software.

Keywords: Collaboration, Framework, IMPCA.

1. Introduction

The purpose of collaboration is “Do the right things and do the things right”. The most important thing to achieve this purpose is to communicate properly and efficiently to each other in organizations. In other words, collaboration is based on communication.

The communication includes both procedural and non-procedural communications. Procedural communication means a thing divided into steps that can be further organized to a procedure such as a standard operation procedure (SOP). This kind of procedures can be handled by workflow management technologies. Non-procedural communication means the communication performed is case by case without a fix form or procedure. This kind of communication can be handled by various communication tools such as telephone, fax, e-mail, instant message, etc. A proper and efficient communication requires three characteristics: correctness, timeliness and safety.

Since collaboration behavior almost implicitly involves a specific communication sequence which is called process, an automatic process control mechanism is required. Workflow management system (WfMS) is introduced to handle this requirement.

Furthermore, the communication produces a great quantity of information, an information analysis mechanism is required for evaluation the rationality of such system. Business intelligence covers this requirement.

This research is to reveal a conceptual framework to suit the needs of a collaboration system, which will cover the process automation topic, and hopefully encourage the integration evolution.

2. Background of Research

Collaborative software, also known as groupware, can be divided into three categories depending on the level of collaboration — communication tools, conferencing tools and collaborative management (coordination) tools. In this research we focus on the collaborative management tools.

Process control is an essential element for the collaborative management system. The development of workflow management systems has been over a decade, there are many issues have been discussed in the evolution. The most important topics for a workflow application framework are security, resource management, process modeling, process verification and simulation, process controlling, and process intelligence. An ideal collaboration system should cover the territories above mentioned to suit the needs of present organizations.

2.1. Security

Security includes network security, identification, access control and cryptography. Currently, the role-based access control is the most popular method. If the security policies can be constructed as well, the access control will be more effective and efficient.

Moreover, the security should be extended to protect the process task data from being amended without authorization. To prevent this from happening, enhancing the access control, intrusion detection and checkpoints are acceptable solutions. The Intrusion Detection System (IDS) could be applied to detect whether the task data are amended or not. If data amended is detected, we can use the checkpoint techniques to rollback the process state to the state just before the data were amended. [14] proposed a solution for on-line attack recovery of workflows.

2.2. Resource Management

For a WfMS, "Scheduling of workflows is a problem of finding a correct execution sequence for the workflow tasks" and "scheduling of a workflow execution involves decisions as to which resources to use and when." [18]. Resource management plays a significant role in this problem, because resources include personnel and physical objects, and some constraints relay to it (time, budget) [18]. Thus, finding an appropriate resource before execute a workflow task is the key to perform a task correctly. "Although resource management has been recognized as an important aspect of a WfMS, most of the work has focused on modeling the various resources with no or little attention devoted to scheduling under the constraints associated with such resources." [18]. [18] proposes a framework (called CCTR) for workflows whose correctness is given by a set of resource allocation constraints and develop techniques for scheduling such systems.

2.3. Process Modeling

The process modeling technology takes the most important role to the WfMS. There are many process modeling related standards existing now such as BPEL, BPML, WSCI, WSFL, XLANG, and XPDL. These modeling languages are briefly described as follow:

BPEL (Business Process Execution Language) is an XML-based language for the formal specification of business processes and business interaction protocols. BPEL extends the Web Services interaction model and enables it to support business transactions [6]. BPEL is currently under standardization at OASIS.

BPML (Business Process Modeling Language) is a meta-language for the modeling of business processes, like XML is a meta-language for the modeling of business data. BPML provides an abstracted execution model for collaborative and transactional business processes based on the concept of a transactional finite-state machine [1].

WSDL (Web Services Definition Language) is a language for describing how to interface with XML-based services. Numerous XML standards have emerged that provide structure to the content exchanged by applications. Such content can be sent and received over a variety of transport protocols. To invoke a service, a consuming application must know the service's interface, including how to structure content and which transport protocol to use. WSDL explicitly describes this interface in a standardized, machine-readable format fit for consumption by tools [19].

WSCI (Web Services Choreography Interface) is an XML-based interface description language that describes the flow of messages exchanged by a Web Service participating in choreographed interactions with other services. WSCI does not address the definition and the implementation of the internal processes that actually drive the message exchange. Rather, the goal of WSCI is to describe the observable behavior of a Web Service by means of a message-flow oriented interface. This description enables developers, architects and tools to describe and compose a global view of the dynamic of the message exchange by understanding the interactions with the Web Service [19]. This standard has been approved by W3C.

XPDL (XML Process Definition Language) is a process description language defined by the Workflow Management Coalition (WfMC). XPDL provides a common method to access and describe workflow definitions. It can act as a template for the creation and control of instances of the process it defines. It can be used for simulation and forecasting and as a basis for monitoring and analyzing instantiated processes. It can also be used for documentation and visualization of the process. XPDL version 2.0 is in draft form and will incorporate BPMN (Business Process Modeling Notation) from BPMI.org. The BPMN specification provides a graphical notation for expressing business processes in a Business Process Diagram (BPD) and provides a binding between the graphical elements and XPDL's block structured elements. This will greatly improve the documentation capabilities for XPDL. Version 2.0 will also provide direct support for message flows which represent communication between processes based on Web Services protocols [17].

BPEL has quickly become the dominant specification to standardize integration logic and process automation between Web services. It has bypassed a number of alternative specifications such as BPML and WSCI, and became a real candidate to drive the existing web service infrastructure into process-centric applications based on a service-oriented architecture [21].

2.4. Process Verification and Simulation

There has been some research on defining formal Petri Net-based execution semantics for business processes. In these approaches, the process definition graph is translated into some form of Petri Net, enabling formal properties such as termination and freedom from deadlock to be proved. Other approaches use formal logic to specify and reason about workflows and transactions. These approaches are focused on verifying process definitions [22].

Once the process definition verification is passed, the process simulation could be performed to simulate the performance of the process. Use the simulation technologies can help people to find the bottleneck of a process. The process simulation generally involves time estimation for every task; and the simulation will use the estimated time requirement for the tasks to be the evaluation basis. Thus, the bottleneck of the process could be predicted.

2.5. Process Intelligence

Persons involved in Business Intelligence (BI) processes may use application software and other technologies to gather, store, analyze, and provide access to data. According to [23], some observers regard BI as the process of enhancing data into information and then into knowledge. The kind of software aims to help people make "better" business decisions by making accurate, current, and relevant information available to them when they need it.

Each business intelligence system has a specific goal, which derives from an organizational goal or from a vision statement. Both short-term goals (such as quarterly numbers to Wall Street) and long term goals (such as shareholder value, target industry share / size, etc) exist [23].

Organizations automate their business processes with the objectives of improving operational efficiency, reducing costs, improving the quality of service offered to their customers, and reducing human error. However, apart from graphical tools for designing business processes, and some simulation tools for detecting performance bottlenecks, business process management systems today provide few, if any, analytic tools to quantify performance against specific business metrics [22].

The emerging area of business process intelligence is aimed at applying data warehousing and data mining technologies to analyzing and optimizing business processes. The log data is typically a basis of the analysis and it stored in a relational database, which can be queried to produce basic reports such as the number of workflow instances executed in a given time period, the average execution time, resource utilization statistics, etc [22].

2.6. Orchestration for the Collaborative Needs

For the collaborative of organizations, the process management architecture of the organizations must provide a mechanism to allow participating organizations to reach agreement (such as ebXML and RosettaNet) on the business process description and the data to be exchanged during process execution. Besides, the process management function needs to be carried out among multiple distributed process managers for the process execution performed by multiple engines collaboratively [22].

Different forms of middleware have been introduced to enable integration and automation of business processes, both within and across organizations. Message brokers, transactional queue managers, and publish/subscribe mechanisms provide means for automating processes by allowing the component applications to post events and to react to events posted by other component applications [22].

3. Proposed Framework

In this research we propose that an ideal Collaboration System Framework is composed of 5 logical layers (IMPCA for the abbreviation): Infrastructure Layer, System Monitoring Layer, Job Planning Layer, Job Controlling Layer and Data Analysis Layer. These layers are composed of many modules as shown in Figure 3.1.

The Infrastructure Layer is composed of Security, System Architecture, Resource Management, Information Exchange Platform, and Application Adapter modules. The purpose of this layer is to build a safe and flexible foundation for system operations.

The System Monitoring Layer includes Monitoring and Recovery, Service Quality Control, and Administration Tasks modules. The purpose of this layer is to keep the system executing normally, and producing management information for the basis of adjusting system.

The Job Planning Layer includes Process Modeling, Process Verification and Simulation,

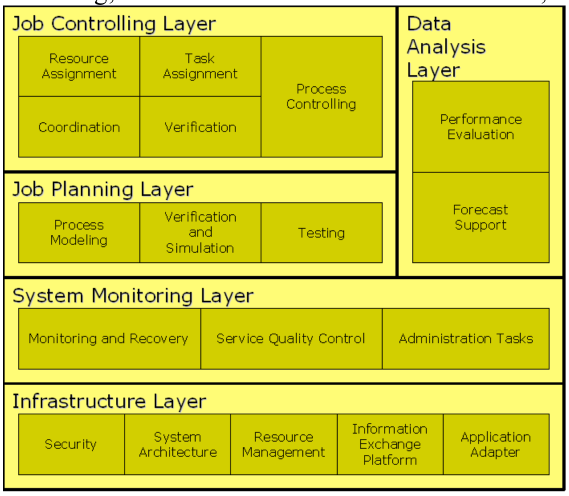

Figure 3.1. The components of IMPCA framework

and Testing modules. The purpose of this layer is to plan the content, pattern, course and goal of a process, and to prove the processes are applicable.

The Job Controlling Layer includes Resource Assignment, Task Assignment, Coordination, Verification, and Process Controlling modules. The purpose of this layer is to control the process execution.

The Data Analysis Layer includes Performance Evaluation and Forecast Support modules. The purpose of this layer is to collect data generated from every layer and transform it into an informative data through data integration, statistics, and analysis techniques; and then provide the informative data to System Monitor Layer as the basis for adjusting system.

The modules of these 5 layers communicate and perform data exchange through the Information Exchange Platform of Infrastructure Layer. The following describes the responsibilities of these 5 layers.

3.1. Infrastructure Layer

Security module includes user authentication, authorization, encryption/decryption mechanisms, and digital signature. The purpose of this module is to provide a consistent safety mechanism.

System Architecture module defines the system architecture which will be applied and constructed in the system (such as standalone, clustering, web based, distributed system, etc.). The mechanism has to be constructed is that it should avoid influencing any other layers or modules while system architecture naturally evolved by the organization's needs.

The Resource Management module takes the responsibilities to allocate, dispatch, and schedule any kind of resources. These resources include personnel (organization structure and personnel), physical/virtual objects, programs, work items, data, events and time. The management of these resources is to control the availability and schedules of personnel and physical/virtual objects. Work item means a task of process defined in a specific process definition. Data include persistence and transient data that may be used by any modules when it required. Event is based on the producer-consumer perspective. An event may not be consumed right away after it generated, instead it may be consumed together with other events when those events arrive. The consumption of events may trigger the generation of a new event, thus, event is also a kind of resource. Time is a resource for scheduling other resources.

Information Exchange Platform provides an information or data exchange mechanism which is controlled and verified by the system. By the use of such mechanism, every module could freely exchange information with other modules and layers. Besides, the information exchange is controlled and verified by the system, thus it could prevent the information exchanged from being polluted by man-made, external components or intrusion. This module covers the territories of data exchange and transmission. Thus this module must provide interfaces of data exchange, transmission and verification for the implementation of variety related adapters.

Application Adapter is an extendable service for adopting user specific services, thus this module must provide interfaces for customized services embedding to the system. From the Model-View-Controller (MVC) architecture perspective, this module is a Controller; the actual functionalities implementation of the extension services (the Model) depends on the requirement and implementation in the designate layer. This module either must be function extendable or conjunction with the Testing module (includes the application life-cycle perspective) of Job Planning Layer to provide a totally integrated environment and support the extendable mechanism.

3.2. System Monitoring Layer

Monitoring and Recovery module takes the responsibilities of maintaining the accuracy, consistency, and availability of data, processes and programs. This module is the controller of recovery procedures. It collects the information of requests and examination reports of every modules, then takes further actions (such as signal a roll-back command) for the related modules (the actual recovery processes performed by the corresponding modules after receiving such commands). This module does not take the responsibilities of the actual examination, verification and recovery procedure, the demarcating of responsibility intends to provide a consistency monitoring and recovery control center. The system could extend unlimited number, type, boundary of monitored objects once it matching the demands of the control center.

The most important job of Service Quality Control module is to construct the service quality policy (Service Level Agreements, SLAs). Once the SLAs is constructed, the numerous system information getting from the system monitors could be mapping and applied to the SLAs to do the further quality control actions (such as feedback to Infrastructure Layer, refine the SOP, etc.). Since the module focus on collecting system behavior information, and such information provides by other modules and layers (mainly comes from the Data Analysis Layer), so that it must define an information collection interface to gather the information provided by other modules.

This module integrates the gathered information and performs further notifications when needed.

Administration Tasks module mainly provides the administrative jobs for maintaining the system operations such as data replication task, job scheduling task.

3.3. Job Planning Layer

The goal of Process Modeling module is to construct a workflow modeling environment, the environment includes modeling languages, modeling tools and data models. In this module, the modeling language plays the most significant role; it determines how to transfer a human perspective workflow to a computer recognized form. The modeling tools provide human friendly tools for the process analyst to perform the process modeling works. The data model indicates the form of computer recognized workflow and the data to be recorded in the computer. It can be divided to three categories: process definition data model (definition data), process data model (runtime data) and application form data model (form data). Process definition data model applies to represents the process descriptions or the result of the process modeling. Process data model applies to represents the run-time data of the workflow execution (such as which task is the current task of the process executed). Application form data model applies to records the data of a workflow application form.

Process definition data model and process data model directly relate to modeling language, because both of these data models are involved in the execution of the process. However, application form data model also has little connection with modeling language, because some information of the application form generally contains useful data for the process execution, such as the different kinds of leave and duration of leave in the leave application form.

Thus, the most important part of Process Modeling module is to decide which modeling language to be used, and then established the whole data model.

Process verification and simulation are based on modeling language, and then refers to the process definition data model and the definition of every token in the modeling language to do the process verification. According to the process verification to proves the accuracy of the process definition. Furthermore, performs process simulation to find out the rationality, bottleneck and estimates the execution performance of the process. The verification functionality of this module is only applies to verifying the process definition.

Software testing is an essential software development requirement for a long time. Integrates Testing module into the IMPCA framework not only treat it as part of software engineering but also considering a whole application life-cycle. Simply, the application life-cycle could be divided to Planning-Implementation-Testing-Online phases. Software testing in every phases of application life-cycle is: establishing test case in the Planning phase, perform unit-test in the Implementation phase, perform integration-test in the Testing phase, but lacks support of run-time testing strategies for the Online phase. Currently, conjunction the bug tracking and bug fixing is the common solution for the system maintenance of online phase. This solution focus on a single problem (or program bug), it shows inadequate while a problem involves many processes. For example: While system runs normally, the system automatically generates a material procurement approval sheet when material inventory is below safety level. Once the system does not generate the approval sheet in a specific situation, it will be hard to realize the problem is causing by receiving material issuance program, inventory checking program or procurement program. Besides, the organization structures are changing rapidly in present days for any kinds of the organizations, to evaluate the existing process definitions could adopting to the changed organization structures or not depends on the online phase testing functionality.

The purpose of IMPCA framework promotes the testing strategy to a global perspective (a system-wide test case) is delegating the run-time testing facility to the system to achieving the whole application life-cycle testability. Once possess the global testing facility, the system self-health-checking could be achieved by scheduling automatic system testing jobs at Administration Tasks module of System Monitoring Layer. Since the system self-health-checking is running on the online environment, so that, we can gather variety online workload analysis information by passing the generated testing data to Data Analysis Layer.

3.4. Job Controlling Layer

The most important thing of Resource Assignment module is to develop resource assignment rules, and the development is based on variety properties of a resource, such as the availability(evaluated by the time resource and the workload of the resource should be considered too, in other word, the quantitative [15]), capabilities, social relationships [15], duties, permissions [4] and intentions(such as a higher priority for a specific person for the intention of experience the on job train) of the resources. According to these properties to develop the rules or criteria of finding and allocating the most appropriate and it's replacement resource from the best suit point of view (such as how to optimize the candidate resources list by a given criteria). For the considerations of the resource assignment, the rules generally combine with the role based resource assignment.

For the Task Assignment module: Tasks are part of the process definition and they define how task instances must be created and assigned during process executions. The most important thing for the task assignment rules is developing the rules of task assignment constrains from the responsibility point of view (such as a task could be acting by someone else or not). For the considerations of the task assignment, it should also consider the problem classification [4] and the qualitative constrains (such as avoid the conflicts to other tasks) of the task relationships [15] for the task to be assigned.

The Coordination module has to detect business process related events based on the agreement of the business process description, and to guarantee reliable delivery of events and messages to applications. Based on the coordination capability, it will be extended to support the recovery of business processes.

The Verification module here is different from the process verification functionality of the Job Planning Layer. The objective of this Verification module is to verify the accuracy and the consistency of the result that comes from the executed process tasks. In other words, this module is deal with the generated information from the processes (the runtime data of processes). The rules for verify the generated information is directly derived from the Coordination module, and it verifies the information exchanged based on the protocol of the Coordination module. The goal of this module is to detect any violation of the data integrity from the exchanged information.

For the Process Controlling, it takes the responsibility of controlling the processes execution. In order to controls the processes execution, there are five parts have to considered: 1. interpreting process definition that created from the Process Modeling module of Job Planning Layer, 2. executing the tasks, 3. transition between tasks based on the transition logics defined by the process definitions, 4. monitoring and invoking the batch jobs, 5. handling the process execution exceptions.

The jobs of interpreting, execution and transition task by the process definition could be done by simply following the definitions of modeling language. For the batch job point of view, a scheduler service is required to invoking the specific job at given time.

For the exception handling, [3] defines an exception as a deviation from the "optimal" (or acceptable) process execution that prevents the delivery of services with the desired (or agreed) quality. This defines a high-level, user oriented notion of exception. From the exception definition point of view, the Service Quality Control module of the System Monitoring Layer will be infected by the output of this module.

At the last of describes this layer, it also has to explain how the Process Controlling module, Task Assignment module, Resource Assignment module, and Resource Management module of the Infrastructure Layer collaborate together.

The Process Controlling module interprets the process definitions, and when the task transition event occurs, it signals a request to the Task Assignment module to do the task assignment jobs. The Task Assignment module identify the request then checks the task assignment rules (such as the acting rule could be differentiated from the application category, for example, a general approval request like Ask Leave Approval could be acting by someone else but a confidential approval request like Witness Protection Plan could not acting by anyone.) that associate to the specific task. Next, the Task Assignment module signals a request to the Resource Assignment module to request an approver candidate list. The Resource Assignment module receives the request and checks the resource assignment rules (such as finds the capabilities requirement criteria for the specific task) that associate to the request. According to the resource assignment criteria ask the Resource Management module of the Infrastructure Layer allocating some candidate resources.

After the Resource Management module returns the candidate resource list, the Resource Assignment module do the optimize filtering by the resource assignment rules to find the optimized approver list for that task. Next, the Resource Assignment module returns the approver list to the Task Assignment module. Finally, the Task Assignment module assigns the task to the chosen approver by the task assignment rules.

3.5. Data Analysis Layer

The Performance Evaluation module takes the responsibility to analysis the collected data that are by the Monitoring and Recovery module of System Monitoring Layer, and transform it to an informative perspective. The On-Line Analytical Processing (OLAP) technology could be helpful in the analysis processes.

Except the needs of performance statistics, analysis and evaluation, the exception analysis also is an important capability of this module due to the exceptions could significantly influence the system performance. The exceptions could be divided to two categories: system exception and resource exception. The system exception means the exception causing by the system error (such as network or program error). The resource exception means the exception causing by a resource limitation (such as a heavy workload causing the job delay). So that, exception analysis helps users in determining the causes of exceptions [3], and users could do the necessary works to prevent the further exceptions occur based on the exception analysis reports.

Forecast support means supporting the prediction of resource usage and exception occurrence. The purpose of predicting the resource usage is to provide an opportunity to prevent exceeds the resource's utilization limitation before exception occurs. In other hand, predicting the exception occurrence is to provide an opportunity to correct or adjust system to avoid violet the SLAs before and after exception occurs. This module allows users and applications to take actions in order to prevent the occurrence of violating the SLAs. For example, the system raises the process priority automatically when this module predicts a process instance has a very high probability of missing its deadline.

In order to support the forecast functionality, the Business Intelligence (BI) technologies could be helpful. From the workflow point of view, "Business process intelligence aims to apply data warehousing, data analysis, and data mining techniques to process execution data, thus enabling the analysis, interpretation, and optimization of business processes" [22]. We could apply the BI to the areas outside the business process control to suit the system-wide needs.

4. An Idea for Implementing the IMPCA Framework

In order to validate the applicability of the IMPCA framework, we survey many open source software and proposing a composition of appropriate modules of them as an implementation of the framework. This conceptual implementation targets at web based application. These chosen open source software are divided to two categories: Development environment tools and IMPCA fundamental tools.

The development environment tools include: Eclipse SDK 3.x (can be downloaded from http://www.eclipse.org/downloads), JBoss Eclipse IDE 1.x (can be downloaded from http://www.jboss.com/), and Eclipse Web Tools Platform 1.x (WTP, can be downloaded from http://download.eclipse.org/webtools/downloads).

The IMPCA fundamental tools include: JBoss Application Server 4.x (JBossAS, can be downloaded from http://www.jboss.com/), jBPM 3.x (can be downloaded from http://www.jboss.com/), Drools rule engine (can be downloaded from http://drools.org/), Bouncy Castle Crypto APIs (can be downloaded from http://www.bouncycastle.org/), Eclipse Test & Performance Tools Platform 4.x (TPTP, can be downloaded from http://www.eclipse.org/tptp), Eclipse Business Intelligence and Reporting Tools 1.x (BIRT, can be downloaded from http://download.eclipse.org/birt/downloads) and Pentaho Business Intelligence 1.x (can be downloaded from http://www.pentaho.org). In addition, a database system would be needed, the MySQL 5.x (can be downloaded from http://dev.mysql.com/) will be used.

J2EE 1.4 specifications defines variety standard services which could be directly applies to the modules of IMPCA framework. Thus, this reference implementation will constructed based on JBossAS, and builds every modules and layers of IMPCA framework in a form of service (web service) upon the JBossAS. In this implementation the JBossAS covers the Infrastructure Layer and System Monitoring Layer. The jBPM covers the Job Planning Layer and Job Controlling Layer. WTP applies to build web applications on the Application Adapter module of Infrastructure Layer. Monitoring Tools of TPTP applies to the Monitoring and Recovery module of System Monitoring Layer. Testing Tools of TPTP applies to the Test module of Job Planning Layer. TPTP, BIRT and Pentaho apply to the Data Analysis Layer. Bouncy Castle Crypto APIs applies to the Security module of Infrastructure Layer. Drools rule engine applies to the Service Quality Control module of System Monitoring Layer and Resource Assignment module, Task Assignment module of Job Controlling Layer. At the last, MySQL applies to the Resource Management module of the Infrastructure Layer to be the main data storage.

4.1. Infrastructure Layer

Since the J2EE 1.4 defined to be an expandable architecture, so that it can natively adopt the system architecture from standalone to distributed, moreover, the applications running on a J2EE compliance application server could adopting to a new architecture without modifications (modify configuration files may be needed) when the system architecture is changed. This expandable architecture drops the influence to the lowest, so that, by using J2EE compliance application server could easily fulfill the requirements of System Architecture module of Infrastructure Layer.

J2EE 1.4 defines numerous security services which contains Java Authentication and Authorization Service (JAAS). In the authentication part, the JAAS supports many authentication mechanisms, such as HTTP Basic authentication, SSL, and HTTP Login Form authentication, even lazy authentication (authenticate the user when a protected resource is accessed) and single sign on both are applicable. In the authorization part, the Java Authorization Service Provider Contract for Containers (JACC) is applied, JACC defines a contract between a J2EE application server and an authorization service provider, allowing custom authorization service providers to be plugged into any J2EE product, so that, any authorization requirements (such as role based authorization, access control list, etc.) could be implemented, and then plug it into security service to accomplish the authorization requirements. Furthermore, for the cryptography mechanism needs could be accomplish by using numerous open source cryptography API (such as Bouncy Castle Crypto APIs), so that develop a cryptography service could fulfill these kind of requirements.

Regarding Resource Management module, JDBC and J2EE Connector Architecture (JCA) could fulfill the needs of accessing the persistence data. JCA is a resource manager integration API whose goal is to standardize access to non-relational resources in the same way the JDBC API standardized access to relational data. In other hand, the restriction rules of accessing transient data are the same as the specifications of any chosen programming language. Controlling the access privilege is the only point which has to be paid attention in this framework.

Each resources of personnel, physical / virtual objects and time must develop dedicated management service, and plug it into JBossAS to be generic resources for any module accessing such resources as the same as persistence data. Finally, the resources of program, work item and event are directly controlled by jBPM.

Regarding Information Exchange Platform, the JBossAS is implemented by using Java Management Extension (JMX) technology. The JMX architecture could be divided to three levels: Instrumentation level, Agents level, Distributed services level. The instrumentation level specifies a notification mechanism. The purpose of the notification mechanism is to allow MBeans to communicate changes with their environment. The Information Exchange Platform can be build based on the notification mechanism to provide the communication capability for every module to communicate with each other. If this module conjunction with Security module, it also can prevent the exchanged information polluted by man-made, external components or intrusion. Moreover, the communication requirements can be accomplished by many existing communication mechanisms, for example, e-mail could be constructed by Java Mail; the point-to-point messaging as well as the publish-subscribe model capabilities (such as instant messaging) could be constructed by Java Message Service (JMS).

Application Adapter module can be easily achieved by the J2EE architecture, because the application adapter facility is a J2EE build-in capability. In other words, Enterprise Java Bean (EJB) technology of J2EE could be applied to extend user specific functionalities. In addition, conjunction with the application server scheduling facility, it could be seamlessly integration with the Testing module of Job Planning Layer.

For the needs of building customized web applications, the Application Adapter module could be a basis of plug-in architecture, and uses WTP to build variety web applications.

4.2. System Monitoring Layer

Regarding recovery facility, the most significant part of it is constructing a Recovery Control Center. It takes the responsibilities of coordination, orders the specific resources to perform the recovery command and reports the recovery execution status, and leaves the actual examination, verification and recovery procedure to the corresponding modules. This service needs to be implemented.

Regarding monitor facility, JBossAS provides many monitor services, and variety monitor related APIs for building user specific monitors. Besides, the Monitoring Tools of TPTP provides frameworks for building monitoring tools by extending the TPTP Platform. In addition, TPTP contains an Agent Controller which enables client applications to launch host processes and interact with agents that coexist within host processes; we can use it to be the basis of monitor controller.

Service Quality Control module mainly constructed into a policy form; so that, it can be constructed by Drools rule engine. Although Service Quality Control module is a form of policies, but its implementation still based on numerous monitors; Data Analysis Layer collects and analyses the information monitors generated, and then verifies it by the quality policy rules; and notifies related persons by the notification mechanism of the Information Exchange Platform when needed. The goal of this module is to monitor the status of services to avoid violet the SLAs.

Regarding administration tasks, JBossAS already provides some administration services and tools. Moreover, the developers could develop customized administration tasks for fulfill the management requirements.

4.3. Job Planning Layer

jBPM is a state machine workflow engine, the jBPM 3.x uses the jBPM Process Definition Language (JPDL) to be the core process modeling language and there is a GUI modeling tool plug-in for Eclipse platform to facilitate the use of JPDL. Besides JPDL, the jBPM 3.x also provides a BPEL extension package that extends jBPM with support for BPEL.

The jBPM 3.x has no process verification and simulation facilities right now. In the fact, there are some open source workflow engine project existing other than jBPM such as Shark (could be downloaded from http://shark.objectweb.org/). The Shark includes a standard implementation completely based on WfMC specifications using XPDL as its native workflow process definition format. Although the Shark provides the simulation ability, it designated to be a standalone product.

Test module is constructed by the Testing Tools of TPTP and scheduler. The Testing Tools of TPTP provides frameworks for building testing tools by extending the TPTP Platform. The framework contains testing editors, deployment and execution of tests, execution environments and associated execution history analysis and reporting. It also includes exemplary tools for JUnit based component testing tool, Web application performance testing tool, and a manual testing tool. The scheduling control could achieve by using Operating System scheduling or J2EE server scheduling.

Regarding Java development environment, the test programs will conjunction with Ant and JUnit. This combination not only fits the principles of light weight software development process, and could easily produces user friendly test reports too. Once conjunction the test programs and build script with scheduling control facility, the developers not only can easily perform the daily test and build in the implementation (development) phase, it also can applies to online testing scenario for the system self-health-checking purpose.

4.4. Job Controlling Layer

Due to the jBPM 3.x does not provides resource management facility, thus it does not provides resource assignment facility too. This jBPM functional limitation is a drawback for standalone applications, but the drawback makes it more suitable for the IMPCA framework, because there is a Resource Management module already existing in the Infrastructure Layer, so that the resource assignment facility could be fulfilled by the Resource Management module, and perform further resource assignment rules development.

jBPM only has a basic task assignment capability, in other words, the capability of mapping task instance and actors. With jBPM, a task instance can be assigned to an actorId, push and pull model of task assignment can be applied in combination. jBPM does not provides the capability of constructing task assignment rules.

Although the jBPM 3.x claims that it provides the orchestration facility. It only means the ability of orchestrate the (web) services but lacks of the coordination functionality. So that the Coordination module has to be implement too. While implementing the Coordination module, the most significant decision is what the protocol (agreement) of the orchestration will be. Once settled the protocol, the development goal of the Coordination module is to ensure the information transportation do follow the protocol specification, and leave the verification jobs to the Verification module. If a slight error or violation from the protocol occurs, a retry action should be taken, and if the violation is critical, this module must signal a recovery request to the Monitoring and Recovery module of the System Monitoring Layer, then the Monitoring and Recovery module signals a recovery command to the relative resources (such as DBMS), and last, the resources execute the process recovery jobs.

Since the Verification module is derived from the Coordination module, so that the Verification module also has to be implemented by self. The goal of this module is to detect any violation of the data integrity, once a failure were detected, this module returns a failure report to the Coordination module, and leave the correction jobs to the Coordination module.

For the Process Controlling module, the jBPM do provides task interpretation, execution and transition functionalities to fulfill the basic process control needs. Besides, the jBPM has a build-in dedicated scheduler service that controls the execution of the scheduled tasks. The most important functions of jBPM are to control and monitor the process execution based on the JPDL process definition that created at the Process Modeling module of Job Planning Layer.

Furthermore, jBPM 3.x has a build-in exception handling mechanism and it is not completely similar to the java exception handling. In jBPM, control flow cannot be changed by the jBPM exception handling mechanism. The exception is either caught or uncaught. Uncaught exceptions are thrown to the client or the exception is caught by a jBPM exception-handler. For caught exceptions, the process graph execution continues as if no exception has occurred.

4.5. Data Analysis Layer

The modules of Data Analysis Layer will be constructed by tools and a few customized services. The tools for this implementation are TPTP, BIRT and Pentaho. The TPTP is playing an evaluated data provider role, the BIRT playing the presentation role for user friendly reporting, and the Pentaho takes the responsibilities of data evaluation and analysis.

The Performance Evaluation module will build based on the TPTP. The TPTP provides many tools for performance evaluation: Profiling Tool, Monitoring Tools and Testing Tools. Plus the Pentaho BI tool, we could easily analyze and present the performance evaluations.

The Profiling Tool of TPTP enables you to profile your applications and resources, and to examine your applications for performance and memory usage problems. The Profiling Tool collects data related to the Java program's run-time behavior and presents this data in both graphical and tabular views. It also enables you to pinpoint the operations that take the most time, and helps you to easily identify repetitive execution behavior and to eliminate redundancy. In short, the Profiling Tool is broadly useful for performance analysis.

The Monitoring Tools of TPTP has the capability of collecting and analyzing system and application resources. The innovative log analysis tools can analyze correlate disparate logs from multiple points across an environment. The Testing Tools of TPTP includes exemplary tools for Web application performance testing tool.

By the use of TPTP, BIRT and Pentaho, users could easily analyze the row data of log and database data. Although the TPTP, BIRT and Pentaho build the fundamental of Performance Evaluation module, some customized evaluation services also needed for the user specific evaluations.

The Forecast Support module provides the prediction functionalities based on the BI technologies. The Performance Evaluation module contributes the knowledge of present and the past status of system usage, and the Forecast Support module will contributes the prediction of the future usage based on the result of Performance Evaluation module. Thus, the goal of the Forecast Support module will achieve by widely using the BI technologies, it means the Pentaho playing the significant role in this module.

In other hand, the customized analysis services have to support the process controlling forecast such as dynamically adjust the task priority at run-time based on the resource workload. The process behavior prediction functionality is not supported by the jBPM 3.x, so that it must be implemented too.

4.6. Summary of the Implementation Idea

In this reference implementation, we integrate many open source products to build the IMPCA framework. The reason of choosing these products to build the framework is these products have been integrated naturally.

Although integrating these products do fulfill the fundamental of IMPCA framework, but it does not covers all the requirements of every modules. These modules must develop numerous services to accomplish the functional requirements. These services include: regarding Infrastructure Layer, the authorization, digest, digital signature, encryption/decryption mechanisms for the Security module; organization and personnel management, physical / virtual objects management, and time management for the Resource Management module; conjunction Information Exchange Platform with Security module of Infrastructure Layer. Regarding System Monitoring Layer, the most important job is developing the SLAs for the Service Quality Control module, then constructing these rules with Drools rule engine and developing a SLAs notification service; in the next, develop a recovery control center and variety monitor services (develop the recovery services in corresponding modules too when needed) in the Monitoring and Recovery module to fulfill the SLAs; besides, develop administration tasks to fulfill the management requirements. Job Planning Layer has to develop the Process Verification and

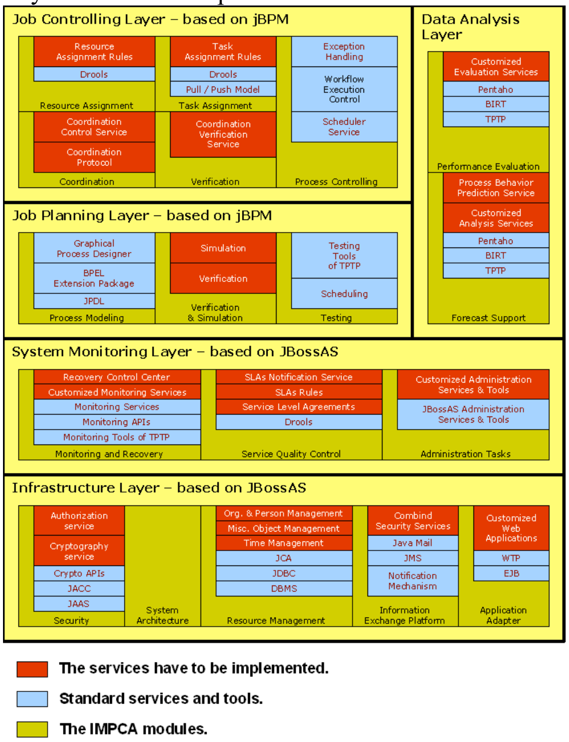

Figure 4.1. The components in the IMPCA

Simulation module. Regarding Job Controlling Layer, the Resource Assignment module must develop resource assignment rules based on the Resource Management module of Infrastructure Layer, and then develop task assignment rules for the Task Assignment module based on the Resource Assignment module. For implement the Coordination module, a negotiation protocol between organizations have to be settled then implement a coordination control service which co-work with Monitoring and Recovery module of the System Monitoring Layer to ensure the process recovery needs, and then construct the Verification module. The Data Analysis Layer has to implement variety evaluation services and analysis services, further more, a process behavior prediction service also needs to implement.

Figure 4.1 shows all of the modules and services combination that builds the whole IMPCA framework.

4.7. Concluding Remarks

An implementation follows IMPCA framework could be a heavy weight task, so that the development of this kind of systems should consider integrating many existing products to reduce development time. For example, the chosen products could be divided to two product families, the Eclipse family for development environment and the JBoss family for IMPCA fundamental (the Pentaho also integrated to this environment since it support Eclipse IDE to be the BI applications development environment and allow deploy applications to J2EE application server), to shows choosing the integrated products could simplify the software integrating effort. Thus, we encourage integrating software to reduce development effort, especially choosing the originally integrated software.

There is an argument of the software choosing. Before we explain the choosing criteria, we will briefly describe the capabilities of the Pentaho first. The Pentaho contains many integrated products include: JBossAS, JBoss Portal, Eclipse BIRT reporting tool, JasperReports reporting tool, Mondrian OLAP server, JPivot, and Enhydra Shark workflow engine to support reporting, analysis, dashboards, data mining, scheduling, web services and workflow facilities.

From the integration point of view, the Pentaho should be the first choice for the IMPCA framework implementation, but we did not. The most significant reason is the process modeling language that we choose. Since the process modeling language is the core element of a workflow engine, the choosing criteria should consider the orchestration needs and long term usability for the collaboration system.

BPEL focuses on issues important in defining web services. It is concerned with message interchange, event handling, locating service instances, transactions, exception handling and compensation. It is suited to being a services orchestration definition language. XPDL focuses on issues relevant to the distribution of work. Activities specify the participants and the applications required to complete the activity. These concepts together allow the system to support the notion of resources which is important for solving business problems [17]. In another words, XPDL focuses on solving the workflow needs.

Since the workflow engine for Pentaho is the Enhydra Shark and the process modeling language of it is the XPDL and has not provided the web service orchestration facility yet. For the considerations of orchestration needs and long term usability, we choose the BPEL to be the process modeling language. The most significant reason for choosing jBPM to be the workflow engine is only because it supporting the BPEL extension. Thus, we use the Pentaho widely except the workflow functionality. This decision propagates to the other software chosen such as the Drools rule engine.

5. Conclusion and Future Works

In this research we propose that an ideal Collaboration System Framework is composed of 5 logical layers: the IMPCA framework. An implementation follows IMPCA framework could be a heavy weight task, so that the development of this kind of systems should consider integrating many existing products to reduce development time. In chapter 4, we present a possibility of integrating many famous open source products to build the reference implementation of the IMPCA framework. Integrating products to build a system could be a short-cut to eliminate development effort. In other hand, instead of developing a brand new system, the developers will spend some time to learn and integrate those products, but the effort should be slighter than developing everything from the beginning. Since the learning and integration time should be considered, the chosen products which have already been integrated with each other will be preferred.

Although integrating many products do reduce a lot of development works, but the implementation shown in chapter 4 still remain a heavy work to do, the customized services, due to the wide scope of the IMPCA framework. The future works should be implementing a real reference application to inspect the IMPCA framework and evaluate the development effort.

REFERENCES

[1] Barry & Associates, Inc., “Business Process Modeling Language (BPML)”, http://www.service-architecture.com/web-services/articles/business_process_modeling_language_bpml.html.

[2] Bill Shannon, “Java 2 Platform Enterprise Edition Specification, v1.4”, http://java.sun.com/j2ee/1.4/download.html, 11/24/2003.

[3] Daniela Grigori, Fabio Casati, Umesh Dayal, Ming-Chien Shan, “Improving Business Process Quality through Exception Understanding, Prediction, and Prevention”, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.

[4] Guido Governatori, Antonino Rotolo, Shazia Sadiq, “A Model of Dynamic Resource Allocation in Workflow Systems”, ADC’2004 Dunedin, New Zealand, Conferences in Research and Practice in Information Technology, Vol. 27.

[5] Håvard D. Jørgensen, “Interaction as a Framework for Flexible Workflow Modelling”, GROUP’01, Sept. 30-Oct. 3, 2001, Boulder, Colorado, USA. ACM 1-58113-294-8/01/0009.

[6] IBM WebSphere software Information Center, “WebSphere Studio tools glossary”, http://publib.boulder.ibm.com/infocenter/adiehelp/index.jsp?topic=/com.ibm.wsinted.glossary.doc/topics/glossary.html.

[7] James Holmes and Herbert Schildt, “Struts: The Complete Referance”, ISBN: 0-07-223131-9.

[8] JBoss, Inc., “The JBoss 4 Application Server Guide”, http://www.jboss.com/products/jbossas/docs, Oct 5, 2005.

[9] JBoss, Inc., “JBoss jBPM User Guide”, http://docs.jboss.com/jbpm/v3/userguide.

[10] John Koenig, “JBOSS jBPM White Paper”, http://www.jboss.com/pdf/jbpm_whitepaper.pdf.

[11] Kirsten Lenz, and Andreas Oberweis, “Modeling Interorganizational Workflows with XML Nets”, Proceedings of the 34th Hawaii International Conference on System Sciences, 0-7695-0981-9/01 2001 IEEE.

[12] M. Millie Kwan and P. R. Balasubramanian, “Dynamic Workflow Management: A Framework for Modeling Workflows”, Proceedings of The Thirtieth Annual Hawwaii International Conference on System Sciences ISBN 0-8186-7862-3/97 1997 IEEE.

[13] Martin Purvis, Maryam Purvis, Selena Lemalu, “A Framework for Distributed Workflow Systems”, Proceedings of the 34th Hawaii International Conference on System Sciences, 0-7695-0981-9/01 2001 IEEE.

[14] Meng Yu, Peng Liu, Wanyu Zang, “Self-Healing Workflow Systems under Attacks”, Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS’04) 1063-6927/04 2004 IEEE.

[15] Minxin Shen, Gwo-Hshiung Tzeng, Duen-Ren Liu, “Multi-Criteria Task Assignment in Workflow Management Systems”, Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS’03), 0-7695-1874-5/03 2002 IEEE.

[16] Nikunj P. Dalal, Manjunath Kamath, William J. Kolarik, Eswar Sivaraman, “Toward an integrated framework for modeling enterprise processes”, COMMUNICATIONS OF THE ACM March 2004/Vol. 47, No. 3, Pages 83-87.

[17] Pentaho Corp., “Workflow FAQ”, http://www.pentaho.org/index.php?option=com_content&task=view&id=62&Itemid=283.

[18] Pinar Senkul, Michael Kifer, Ismail H. Toroslu, “A Logical Framework for Scheduling Workflows under Resource Allocation Constraints”, Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002.

[19] Reactivity.com, “XML Web Services and SOA Guidebook: Glossary”, http://www.reactivity.com/guidebook/glossary.html.

[20] Shazia Sadiq, Maria Orlowska, Wasim Sadiq, Cameron Foulger, “Data Flow and Validation in Workflow Modelling”, ADC’2004 Dunedin, New Zealand, Conferences in Research and Practice in Information Technology, Vol. 27.

[21] Tom Baeyens, “The State of Workflow”, http://www.jboss.com/products/jbpm/stateofworkflow, October, 2004.

[22] Umeshwar Dayal, Meichun Hsu, Rivka Ladin, “Business Process Coordination: State of the Art, Trends, and Open Issues”, Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.

[23] WIKIPEDIA, The Free Encyclopedia, http://en.wikipedia.org/wiki/Main_Page.

[24] Workflow Management Coalitition, “The Workflow Reference Model”, Document No. TC00-1003, Issue 1.1, 1995.