國立交通大學

資訊工程學系

博士論文

Agentflow - 一協同合作跨組織的工作流程系統

Agentflow - A Cooperative Inter-organizational Workflow System

研 究 生：殷玄彬

指導教授：王豐堅 教授

中 華 民 國 九 十 四 年 六 月

Agentflow - 一協同合作跨組織的工作流程系統
Agentflow - A Cooperative Inter-organizational Workflow System

研 究 生：殷玄彬　　Student：Shung-Bin Yan

指導教授：王豐堅 教授　　Advisor：Feng-Jian Wang

國立交通大學 電機資訊學院

資訊工程學系

博士論文

A Dissertation

Submitted to Department of Computer Science and Information Engineering

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

Hsinchu, Taiwan, Republic of China

June 2005

中華民國九十四年六月

博碩士論文授權書

本授權書所授權之論文為本人在 國立交通 大學(學院) 資訊工程學 系所
組 93 學年度第 2 學期取得 博 學位之論文。
論文名稱： Agentflow - 一協同合作跨組織的工作流程系統
指導教授： 王豐堅 博士

1.■同意 □不同意

本人具有著作財產權之上列論文全文(含摘要)資料，授予行政院國家科學委員會科學技術資料中心(或改制後之機構)，得不限地域、時間與次數以微縮、光碟或數位化等各種方式重製後散布發行或上載網路。
本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為:_____，註明文號者請將全文資料延後半年再公開。

2.■同意 □不同意

本人具有著作財產權之上列論文全文(含摘要)資料，授予教育部指定送繳之圖書館及國立交通大學圖書館，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會及學術研究之目的，教育部指定送繳之圖書館及國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，不限地域與時間，讀者得進行閱覽或列印。
本論文為本人向經濟部智慧財產局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為:_____，註明文號者請將全文資料延後半年再公開。

3.■同意 □不同意

本人具有著作財產權之上列論文全文(含摘要)，授予國立交通大學與台灣聯合大學系統圖書館，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會及學術研究之目的，國立交通大學圖書館及台灣聯合大學系統圖書館得不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。
論文全文上載網路公開之範圍及時間－
本校及台灣聯合大學系統區域網路： 97 年 6 月 2 日公開
校外網際網路： 99 年 6 月 2 日公開

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鉤選，本人同意視同授權。

研究生簽名： 學號：8317504

（親筆正楷） （務必填寫）

日期：民國 94 年 6 月 3 日

---

1. 本授權書請以黑筆撰寫並影印裝訂於書名頁之次頁。

國家圖書館博碩士論文電子檔案上網授權書

本授權書所授權之論文為本人在 國立交通 大學(學院) 資訊工程學 系所
組 93 學年度第 2 學期取得 博 士學位之論文。
論文名稱： Agentflow - 一協同合作跨組織的工作流程系統

指導教授： 王豐堅 博士

■同意 □不同意

本人具有著作財產權之上列論文全文(含摘要)，以非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未鉤選，本人同意視同授權。

研究生簽名： 學號：8317504

（親筆正楷） （務必填寫）

日期：民國 93 年 6 月 3 日

---

1. 本授權書請以黑筆撰寫，並列印二份，其中一份影印裝訂於附錄三之一(博碩士論文授權書)之次頁；另一份於辦理離校時繳交給系所助理，由圖書館彙總寄交國家圖書館。

Agentflow - 一協同合作跨組織的工作流程系統

學生：殷玄彬　　指導教授：王豐堅 博士

國立交通大學資訊工程學系（研究所）博士班

摘　　要

工作流程管理系統(WfMSs)模型化和管控商業流程(business process)的能力已全世界被接受。過去，我們定義一個組織內部工作流程模型(PLAN)並實作相關的工具，允許使用者可以用圖型化方式描述流程，表單與組織架構。 PLAN 已成功在 Agentflow 裡實現並用以執行工作流程應用程式。 PLAN 和大部分目前的工作管理系統是採用集中化的架構只適用於單一組織內部使用，而不同組織之間的流程應用程式很難達成。本論文中提出一個以服務為導向之工作流程模式(CA-PLAN)用以支援跨組織的商業流程功能。在 CA-PLAN 中，一個跨越組織的流程依據組織之界線劃分成多個單一組織內之流程。在每一個組織內的工作流程系統被視為一個工作流程整合元件(IWC)。每一個 IWC 包含一個流程服務介面，遠端流程介面和組織內部的流程。一個跨組織之流程是經由相關聯 IWC 的流程服務與遠端流程之間連接做成。而且，在 Agentflow 系統(架構和實作)裡提供的相關服務技術和支援之工具也被提出。

除此之外，我們也提出一聰明且個人化的企業流程入口網站架構。 工作流程應用的腳本不再局限於小的用戶組織操作文件管理或者數據處理應用程式，商業流程自動化和整合變得更複雜。透過企業流程入口網站與軟體代理人機制來佈署流程應用程式用以幫助員工的每日作業和經理在執行決策時之建議。

Agentflow - A Cooperative Inter-organizational Workflow System

Student：Shung-Bin Yan Advisors：Dr. Feng-Jian Wang

Department of Computer Science and Information Engineering

National Chiao Tung University

Abstract

Workflow management systems (WfMSs) are accepted worldwide due to their ability of modeling and controlling business processes. In the past, we defined an intra-organizational workflow specification model, Process LANguage PLAN. PLAN, with the associated tools, allows user to describe the graph specification for processes, artifacts and participants' organization. PLAN has been successfully implemented in Agentflow to support workflow (Agentflow) applications. PLAN and most current WfMSs are designed to adopt the centralized architecture such that they can be applied in an organization only. However, the participants of Agentflow applications in different organizations can hardly serve each other with workflows. In this dissertation, a service-oriented cooperative workflow model, Cooperative Agentflow Process LANguage (CA-PLAN) is presented. CA-PLAN proposes a workflow component model to model inter-organizational process. In CA-PLAN, an inter-organizational process is partitioned into several intra-organizational processes. Each workflow system inside an organization is modeled as an Integrated Workflow Component (IWC). Each IWC contains a process service interface specifying the process services provided by the organization, in conjunction with a remote process interface specifying what the remote processes are used to refer to remote process services provided by other organizations, and intra-organizational processes. IWC is a workflow node and participant.

An inter-organizational process is made of connection among these process services and remote processes with respect to different IWCs. Besides, the related service techniques and supporting tools provided in Agentflow system (architecture and implementation) are also presented.

Besides, we also present an intelligent and personalized enterprise process portal. The scenarios of workflow applications are no longer limited to small user groups operating the document management or data processing applications. Meanwhile, business process automation and integration are getting more sophisticated. The system deploys workflow applications inside the Enterprise Process Portal to help employee's assistants in daily works as well as manager's advisor in executive decision.

Keywords: service-oriented workflow, integrated workflow component, workflow management system, software agents, enterprise process portal, reusability.

誌謝

本論文終於完成，首先感激我的指導教授王豐堅老師，假如沒有他的殷勤教誨與嚴格督促，這本論文將無法在此刻順利完成。

其次我要感謝所有的口試委員：朱治平教授、葉義雄教授、陳耀宗教授、朱正忠教授、楊鎮華教授及留忠賢教授給我許多寶貴的建議與指正，使得這篇論文更加詳盡完美。另外，我要謝謝曾經提供我在研究及各方面協助的實驗室學長學弟：梁賓先，楊基載，陳英信，邱威傑，王志誠，林祖年，許薰任等以及華苓公司的夥伴們魏禎德、黃齡慶、張俊謙、饒彰年、蔡慶鴻等在論文寫作這段期間的幫忙。還有許多曾經協助我的師長朋友，未能全部列出，仍要在此感謝。生命中能和這些敬愛的師長和可愛的朋友相遇，真是我的福氣。特別感恩我的父親及母親過去辛勤的養育和教導。誠摯地以此研究成果獻給我的家人。

殷玄彬

九十四年仲夏於新竹

目　錄

表 目 錄

圖 目 錄

Chapter 1 Introduction

1.1 Motive

In the eBusiness age, business environments have become exceedingly dynamic and competitive. For customer's requests, efficient, effective, and accurate response is vital for success of an enterprise. In most organizations, requirement changes in business processes occur frequently and prevalently. The enterprises need to better operations in a state of perpetual change and adaptation. To address these requirements, enterprises must constantly reconsider and optimize how to support evolving business process [1].

Traditionally, enterprises use a bottom up and data-centric approach to construct their information system. The logic of process, application, and organization are combined together. When a process is tightly coupled in application codes (when built from scratch) it is inaccessible and difficult to change. As a result, the cost of maintenance is high. For a manager of information system, it is a nightmare to update a system due to an additional requirements or changes in the business rules or process. Workflow is a technology that allows users to create a process layer, which provides a level of process abstraction, and removes the processes from the control of applications. It separates the process logic, application logic, and organization logic. The process is maintained independently of the core functionality of the application, the cost of maintenance is considerably lower. The information system that constructs by workflow technologies (top-down, process-centric approach) not only manages business data but also business process (people-intensive applications) and process data that provide opportunity to optimize the process.

An enterprise needs to construct a process-centric organization and information system for supporting rapid changes. Workflow can be defined as the automation of a business process during which documents, information or tasks are worked and passed from

and to other processes according to a set of procedural rules [2]. Workflow technology improves an organization's performance in doing planned work.

Workflow management systems (WfMSs) are accepted worldwide due to their ability of modeling and controlling business processes. WfMS can effectively manage, dispatch, and coordinate activities of people working on a common task or project. WfMS ensures that the right information reaches the right person or computer application at the right time. Successful WfMS deployment results in significant process cycle time reductions, cost reductions, improved accuracy, greater control, and greater worker satisfaction. A workflow has been expected to spread across company boundaries on the Internet. Therefore, a technology for inter-organizational workflow becomes an important issue. Recent researches in WfMSs have been focusing on cooperative e-Business among different organizations that use their individual resources to achieve a common goal. These individual resources contain application services, business processes, and people, etc. Hence the requirements of inter-organizational workflow have emerged in medium-to-large-sized enterprises. A conventional (called intra-organizational) process is a process whose activities are all performed within the same WfMS of a single organization. An inter-organizational process contains one or more cross-organization sub-workflow (called co-process in different sites.) where each contains three process elements (process service, remote process and process for the cooperation workflows in different organizations. An inter-organizational WfMS is extended from the traditional WfMS to allow user to model inter-organizational processes and provide a workflow engine to enact, manage and monitor the execution of inter-organizational processes.

Because of the dynamic nature of e-Business, the inter-organizational WfMS must be able to accommodate the participating organizations to change their business processes and strategies dynamically, to handle events whether they are predefined or not, and to support run-time modifications of process models [3]. *Task monitor* is also an important tool for

WfMS to track and monitor the status and the progress of each enacted process instance. Through task monitor, workflow participants and administrator can know each instance‘s current status and the progress of an enacted process. However, security issues between different inter-organizational WfMSs must be considered for enacting an inter-organizational process across organizational boundaries. The inter-organizational process monitor and administration must be configured with appropriate authorities for different workflow applications.

1.2 Cooperative Agentflow Process LANguage

Cooperative Agentflow Process LANguage (CA-PLAN) is designed to extend the underlying model of Process modeling LANguage (PLAN) [4] by adding remote processes, process services, and process control data as its modeling constructs, and by allowing a process in one WfMS to enact a remote process in another WfMS over network. An inter-organizational process is a process whose activities are performed to cooperate, coordinate, and interact among the processes in different organizations over the Internet. In CA-PLAN, an inter-organizational process could be partitioned into several parts which have boundary functionalities in different organizations. CA-PLAN lets a workflow system inside an organization be modeled as an Integrated Workflow Component (IWC). Each IWC contains a process service interface specifying the process services provided by the organization, to be coupled with a remote process interface specifying the remote processes used in the intra-organization processes, and intra-organizational processes of the organization. Process services and remote processes are input and output connector pins of an IWC. A co- process is defined as a process which contains a pair of process service and remote process supported by IWC connection between their process services and remote processes.

There are three layers for presentation and detailed handlings for an inter-organizational process. The top layer lets designer design an inter-organizational process as an

intra-organizational process (i.e., processes in WfMS). The medium layer, based on Java machine, provides a communication and management mechanism to help the design implicitly. This cooperative mechanism between WfMSs in CA-PLAN is modeled as a process-specific Remote Call Process (RCP) mechanism of which it contains implicit input and output connectors with an intra-organizational process component. RCP also supports the ability to manage, control and monitor the execution of an inter-organizational process instance. Also, the bottom layer is a transport layer that may use http, web service, java RMI …etc.

1.3 Enterprise process portal

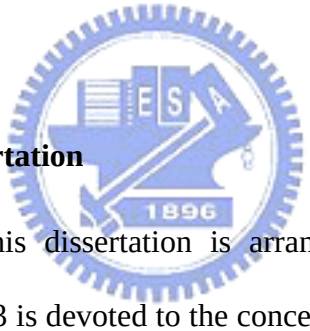
The research of software agents [5][6] during last decade has accumulated lots of outcome and demonstrated theirs applications in different areas. Software agents have several useful characteristics such as mobility, autonomy, and intelligence. They are also proactive to their executing environments. Therefore, researchers and software industry in workflow technologies, start to adopt software agents as part of the software design by viewing them as autonomous and mobile software components.

Our work only focuses on the enrichment of software agents that are directly related to the workflow platform and workspace. The enrichment allows software agents in the workflow system to make efforts on monitoring the operation workflow engine and the interaction of workflow participants. Software agents employed need more intelligence than mobility. Here, most of the software agents mentioned in the article are *stationary* agents (that never migrate), i.e., the agents do move on the intent sites.

The enterprise process portal (EPP) is a portal that provides user interfaces for general workflow operations, and augments traditional portal applications with relevant process information. With the ability of modular and personalized page composition for end-user screen, our workflow system can provide intelligence on the agents based on the end-user's

preference. A sample software agent is the decision analysis agent that analyzes the user's decision history of the same type, and takes current application data as input parameters (e.g., the payment amount and the position of applicants in the payment approval process) to suggest the decision on the application screen, and/or make the decision as the default action when the end-user intends to submit the activity and continue next step without any comments.

The software agents of that class can be added incrementally to the corresponding enterprise process portal by changing the configuration of the portal, because the independent suggestions of software agents can be displayed in their portlets of the Web page. The arrangement reduces the required efforts for the business process consultants to inject their ideas into the software agent's knowledge. Thus, the workspace containing injected business process intelligence can help users to improve their efficiency. The process portal can be extended with more portlets.



1.4 Organization of this dissertation

The organization of this dissertation is arranged as follows. Chapter 2 presents workflow and PLAN. Chapter 3 is devoted to the concept of the cooperation workflow Model CA-PLAN. Chapter 4 describes the Agentflow architecture and design. Chapter 5 describes the implementation and related issues of a cooperative Agentflow. Chapter 6 presents an intelligent and personalized enterprise process portal. Conclusions and our future work are finally covered in Chapter 7.

Chapter 2 Related Work And Background

2.1 Workflow background information

In this session, we give a simple introduction on the workflow and workflow management system.

2.1.1 Workflow and Workflow Management System

Workflows have been introduced in many enterprises for the purpose of automation and improvement of the business process. In this dissertation, we use terminologies in the Workflow Management Coalition (WfMC) [2]: workflow is "the 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"[7].

WfMS is a kind of software application that supports the specifications and the executions of workflow. WfMS provides the "procedural automation of a business process by management of sequence of work activities and the invocation of appropriate human and/or IT resources associated with various activity steps" [7]. It contains a set of tools for the support of necessary services, such as workflow creation, workflow enactment, and administration and monitoring of workflow processes. A WfMS provide supports in three main function areas [8]:

- The build-time functions, concerned with defining and modeling of the workflow process and its constituent activities.
- The run-time control functions, concerned with managing the workflow processes in an operational environment and sequencing the various activities to be handled as part of each process.
- The run-time interactions, with human users and IT application tools, for processing the various activity steps.

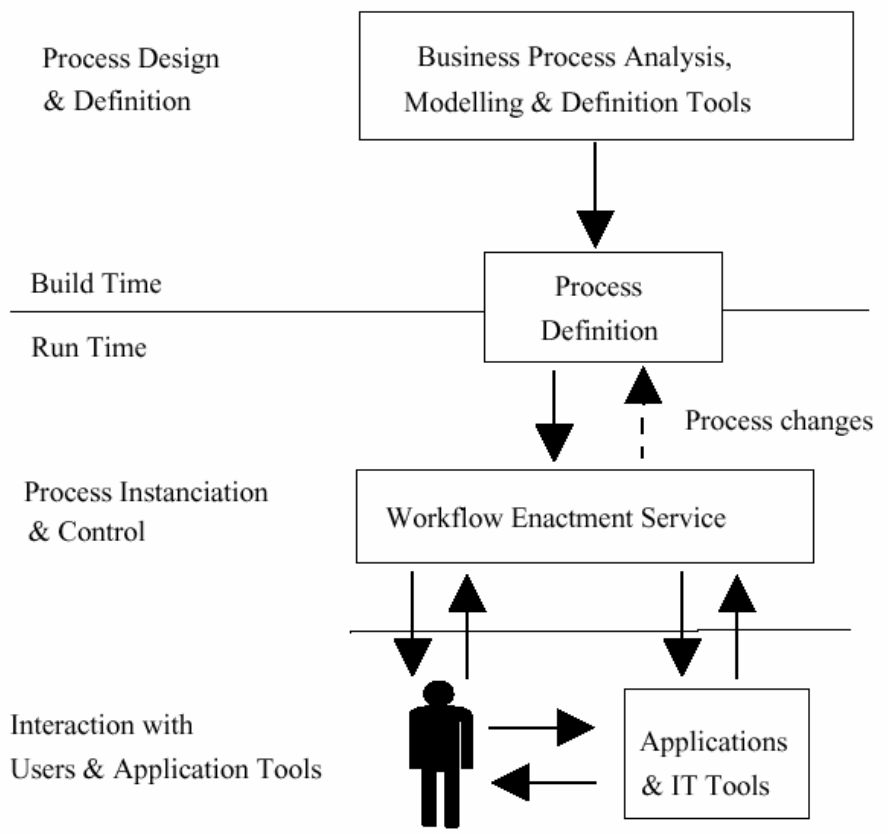


Figure 1. Workflow system characteristics

Figure 1 illustrates the basic characteristics of WFMS and the relationships among these main functions.

On the other hand, a workflow application can be divided into two main parts:

- ✧ Workflow process design and definition in the build-time.
- ✧ Workflow process instantiation and control during the run-time.

Since WfMS is flexible and easy to build a workflow application, a process designer only pays his/her attention on solving business logic (high-level design) of enterprise without concerning hard-code (low-level design) of common services, such as “how to transmitting the message by network to my manager, “how to enact the rule of dispatching task in one hour”. WFMS will be in charge of all general services related to workflow process, e.g. creating, dispatching, and altering works, etc.

2.1.2 The Workflow Reference Model

WfMC is a non-profit organization whose mission is to establish standards for workflow terminology, interoperability and connectivity [2]. The *Workflow Reference Model* [8] proposed by WfMC**,** as illustrated in Figure 2, defines several components and interfaces. The interface around the workflow enactment service is designated as *Workflow APIs* (WAPI) and *Interchange formats*, which may be considered as a set of constructs by which the services of the workflow system may be accessed and which regulate the interactions between the workflow control software and other system components.

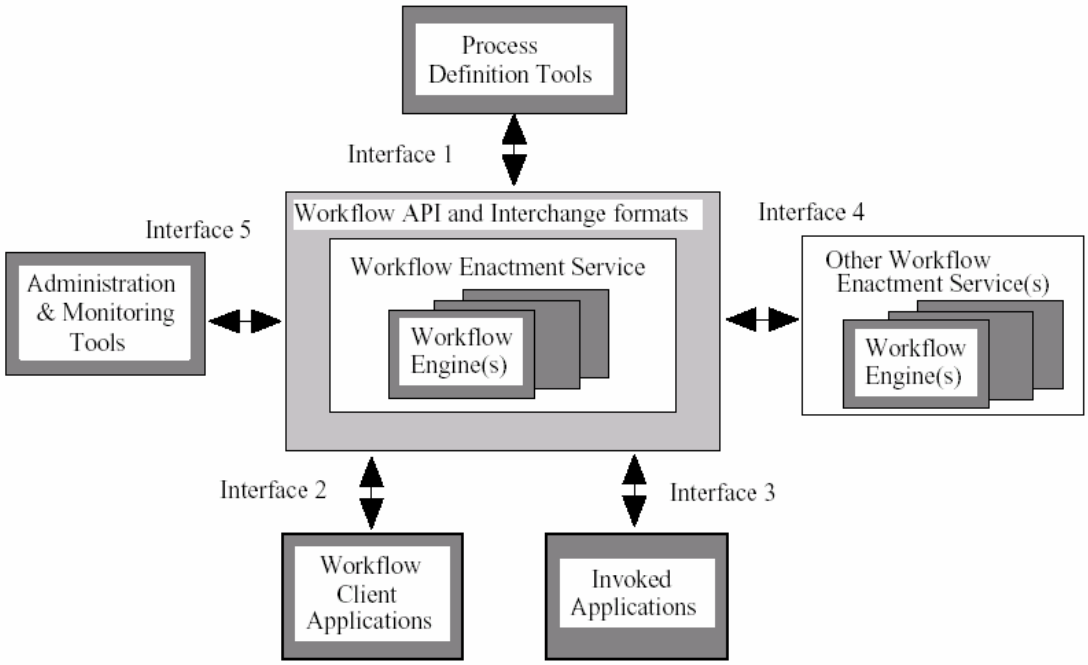


Figure 2. WfMC's Workflow Reference Model

2.2 Past researches

There are several WfMSs designed and implemented to support interoperability[3][9][10][11][12][13][14][15][16][17][19][20][21]. Events and rules have been used in several workflow projects. Fabio Casati [11] [12] presents a workflow event model, of which it is a message passing based model, and it allows interoperation among processes in different organizations in the WIDE [16] project. In his work, the traditional

workflow model by enhancing the traditional workflow to publish and subscribe to event for interaction was extended. The event dispatching mechanism that he proposed handles the events according to the priorities of event publication, business convention or cost…etc. The METEOR [17] project of Georgia University uses the interface-based generator to produce the interoperable interfaces during design time. When two WfMSs need to be interactive with each other, they call interfaces via CORBA-IIOP [18]. Stanley YW Su [3][13] presents a solution to achieve dynamic Inter-enterprise workflow management using the e-services provided by cooperative e-business enterprises. This constraint-based dynamic e-service binding mechanism allows inter-enterprise workflow process models to be processed in the Internet environment, in which business organizations and their services are changing constantly. Davulcu [19][20] proposes a framework based on Concurrent Transaction Logic (CTR) for modeling and reasoning about interactions in a virtual enterprise. He uses CTR to coordinate execution sequence of processes. However, it is difficult for user to construct workflow processes. Dickson [21] proposes a meta-model of workflow views and develops an interoperation model based on workflow views. However, each party may need to make modification to its own workflow for interoperable interface. The interoperation mechanism is not loosely coupled; in other words, a party that modifies its own workflow may influence other party's workflows or views.

The Workflow Management Coalition (WfMC) [2] has also proposed an abstract interoperability specification to allow the interoperability of workflow specifications supported by different WfMSs. It releases Wf-XML [22] specification of standard message format for interoperation. It provides a structure and well-defined model for WfMSs interoperability, which is independent from the transport mechanism and platform. These standard specifications are summarized and used to propose a new cooperative workflow meta-model to define cooperative processes between different WfMSs. Projects mentioned above extend the WfMC's WPDL model by adding different constructs to model the inter-

organizational process. Their works primarily focus on transport mechanism of inter-organizational process.

Based on organizational viewpoint, it is necessary to decompose an inter-organizational process and propose an integrated workflow component model to describe it. The security issues of enacting and monitoring an inter-organizational process between different WfMSs across organizational boundaries should be considered for different workflow applications. The inter-organizational process monitor and administration between different WfMSs should be configured with appropriate authority for distinct workflow application.

2.3 A Simple overview of PLAN

The Software Process Model, PLAN (Process LANguage) [4], developed in our laboratory, is used to model the software development project. PLAN was built to write the context that indicates how activities will be performed. This in turn manipulates artifacts with the required resources, as shown in Figure 3. A role, such as programmer, defines the responsibilities within an organization. Besides, a role has administrative relationships with sub-roles. To do a project, each role needs to communicate with other roles. A process can also be decomposed into a set of sub-processes/activities. There are two kinds of activities: operation and analysis. An operation modifies the state of an artifact(s). To gauge project information, an analysis refers to, but not the changes of artifact states. Process behavior controls the execution sequences of activities. Each artifact has its own lifecycle, which is characterized by states during development. Furthermore, an artifact may consist of a set of sub-artifacts. As depicted in the conceptual framework, a PLAN process definition also contains the web-linking semantics of a software process.

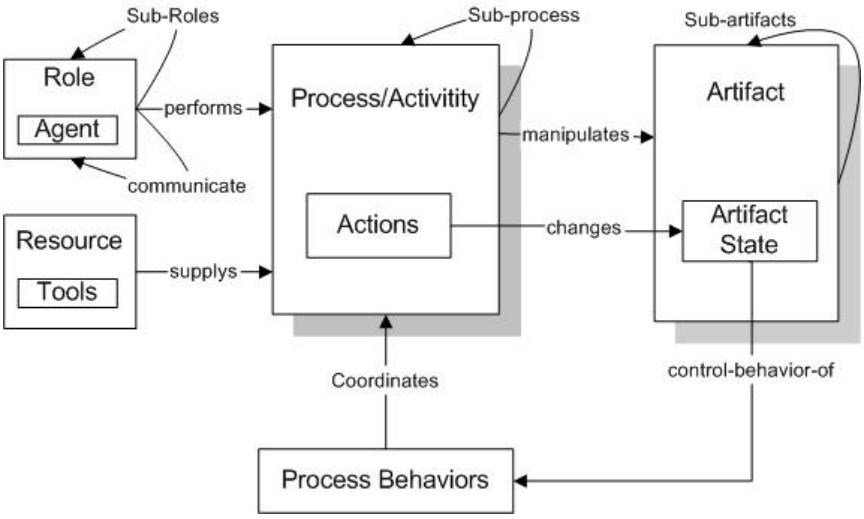


Figure 3. Conceptual Framework of PLAN

Chapter 3 Workflow Development Environment - Agentflow

3.1 Cooperation Workflow Model CA-PLAN

The main issue for a workflow system is answering the question “who must do what, when and how”. Workflow is defined by WfMC [2] as the 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 [1]. According to the definition, a process consists of three major elements as shown in Figure 4.

- Activity – activities involved in the processes coordinate according to a set of procedural rules. An activity (the “what” part of the issue) presents something to be done and transitions define the appropriate sequence of activities of a process (the “when” part of issue).
- Artifact – documents, information handled by the processes. The “how” part of the issues specifies an associated application of each activity.
- Role – the roles of the participants involved in the processes. The “who” part of the issue specifies the participants of each activity.

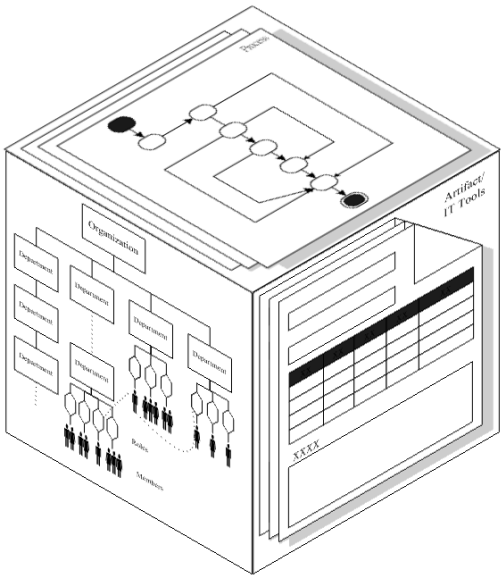


Figure 4. Three major elements of a process: activities, roles, and artifacts

3.1.1 CA-PLAN Meta Model

The workflow meta-model describes the top-level entities contained within a workflow process definition, their relationships and attributes. Cooperative Agentflow Process LANguage (CA-PLAN) meta-model as shown in Figure 5 supports general workflow processes, and simplifies seamless remote process invocation to suit for inter-organizational workflow.

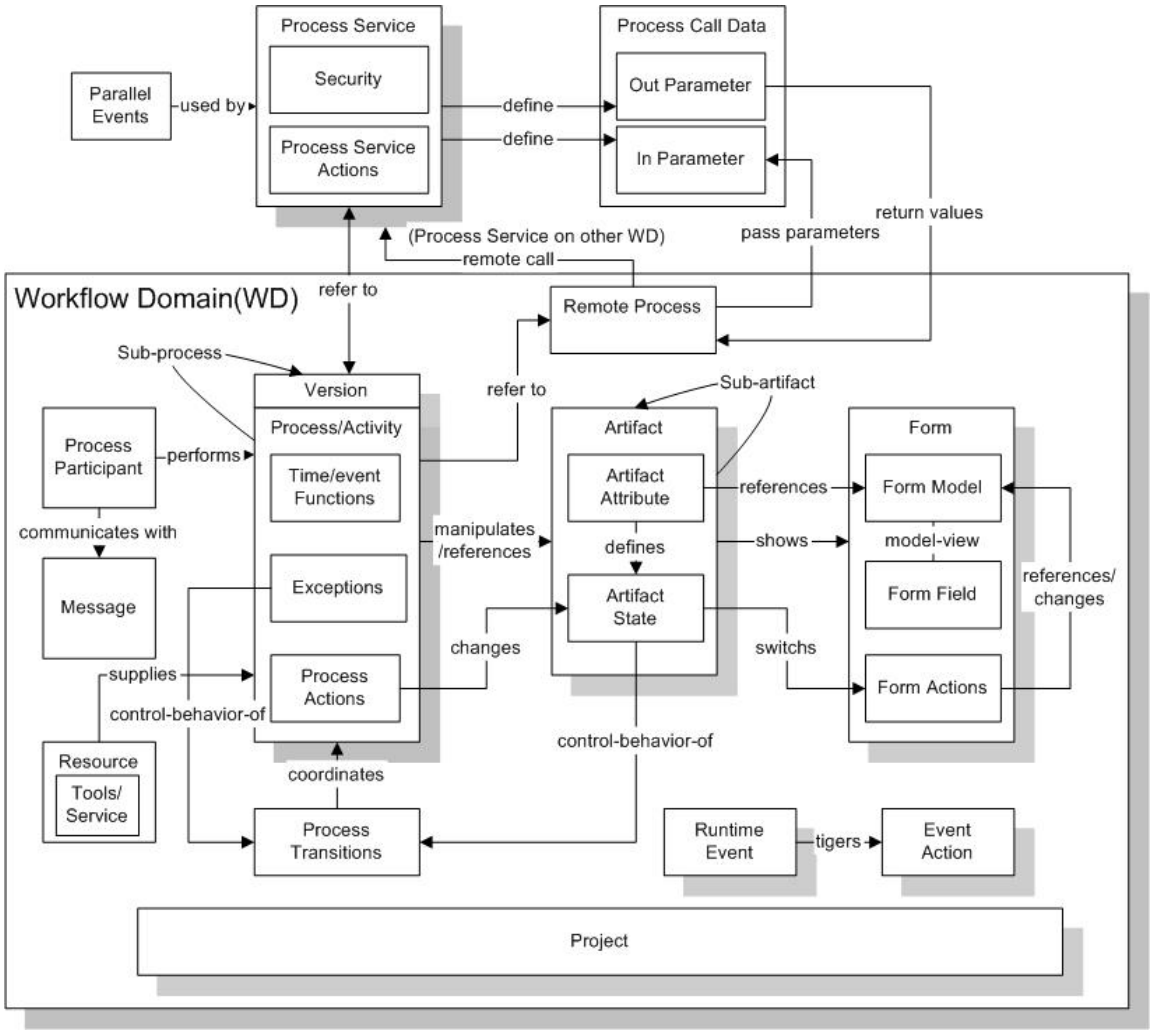


Figure 5. The CA-PLAN Meta Model

A Project is a collection of related processes, artifacts and project roles providing access protection and namespace management. A Process can be decomposed into a set of sub-processes/activities. An activity can be divided into two types: automatic activity and human activity. A human activity is assigned to Process Participants to perform the work and

an automatic activity is automatic execution by computer applications. An Activity also contains entrance condition, exit condition, version, time/event functions, exceptions, and process actions (Pre-Action, Action, Post-Action) attributes.

Artifact has two elementary properties Artifact States and Artifact Attributes. Artifact Attributes define the data model of workflow-application that handled by a process. Each Artifact has its own lifecycle, which are characterized by Artifact States during development. Each Artifact State contains a state condition that is a logic expression defined by Artifact Attributes.

An Activity may manipulate an Artifact and change the state of the Artifact. The Process Transitions define the execution sequences of Activities. Each individual Process Transition has three elementary properties, the from-activity, the to-activity and the conditions (involving expressions which are evaluated to permit or inhibit the transition) under which the transition is made. The Transition condition is an expression by an Artifact State of the operation Artifact of the form-activity. The transaction is fired when the from-activity manipulates the attributes of the operational artifact and changes the artifact's state that satisfies the condition of the transaction.

An Artifact can view as a workflow-application data model and have many different types of views (Form). A Form can have many types presentation (HTML, Applet, ActiveX, PDF form, Word, etc…). A Form contains a set of Form Fields and Form Actions. An electronic Form is a container that can contain all of the software components (label, button, text field etc…) in the window's GUI. The Form Field specifies the attributes of the software components. Form Actions (PreAction, OpenFormAction, CloseFormAction, PostAction, and Corresponding Actions of Software Components) define corresponding operations that are triggered when users manipulate the Form.

Each Process Participants can communicate with other Process Participants by Messages. The Process Participant provides descriptions of resources that can act as the

performer of the various activities in the process definition. The Process Participant assignment mechanism has two binding types: Design-Time-Binding or Run-Time-Binding. The Design-Time-Binding mechanism selects the candidates for the Process Participant in the process design time phase. However, Run-Time-Binding selects the candidates for the Process Participant to delay in the process enact phase. The Process Participant declaration may refer to organization model, project role model, process performer of another Activity, Artifact Field (that contains user information) or user customized method that implemented by scripts of process actions, as shown in Figure 6.

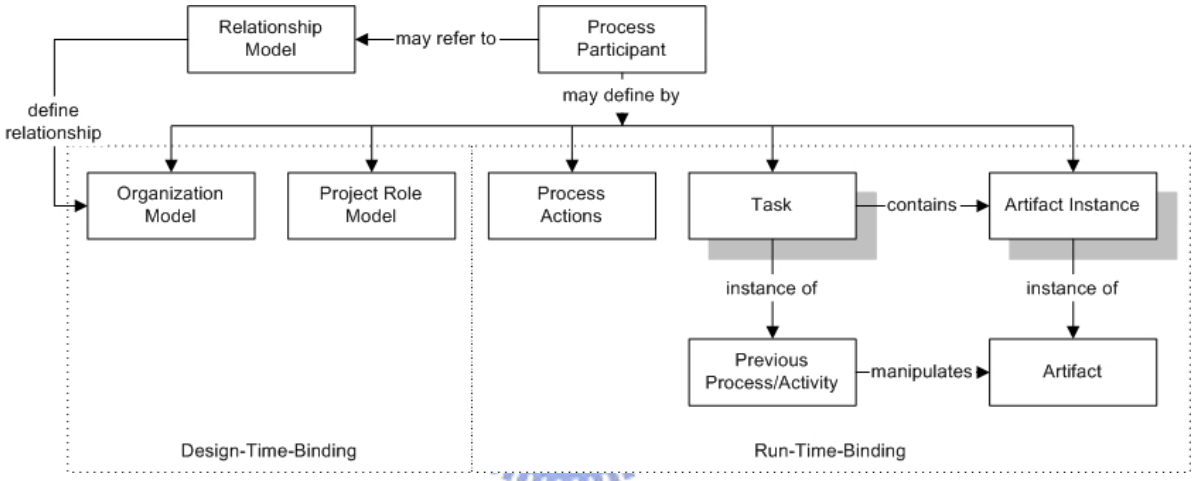


Figure 6. Process participant assignment mechanism

Agentflow provides an event-action handling paradigm for user to interact with runtime operations of a WfMS. The workflow runtime system may generate many kinds of Runtime Events, for example ServerStart (workflow engine starting), ServerStop (workflow engine closed), RedispathTask (administrator reassign a task), DeputyEnable (user enable the deputy mechanism) … etc. Event Actions can be registered as event listener on a Runtime Event. If the Runtime Event is fired by system, the corresponding registered Event Actions will be triggered and executed. The Process Action, Form Action, and Event Actions use the Agentflow scripting language that the syntax is same as JavaScript [23] and extends some Agentflow host objects) to manipulate, customize and automate the facilities of Agentflow workflow system.

To adapt dynamic and competitive business environment, enterprises constantly reconsider and optimize the way they do business and change their information system and applications to support evolving business processes. Agentflow system provides a process version mechanism to support process evolution. In the mechanism, if a process definition needs to change, the reviser creates a new definition version from current version and modifies the created one.

To support process design with better modularity and provide process a reuse mechanism, Agentflow system provides a Call-Process activity construct. A Call-Process activity is an activity that contains the attributes including a called process's ID and a unique identity. The called process's ID refers to an ID of the called process. The definition of a called process may evolve to a new version; if no specification, the Call-Process will use the definition of newest version for the enacted process (instance) process to execute. This version mechanism allows user to modify the definition of a called process with different versions, and all Call-Processes will take the changes.

A Workflow Domain (WD) is a workflow management boundary that provides a common Process Definition Repository (PDR) and Runtime Repository (RTR). The PDR stores process definitions including projects, processes, artifacts, and forms. The RTR stores execution and log data of processes including process instances, artifact instances, and form application data. A workflow domain may consist of one or more workflow engines that access the same PDR and RTR. An organization can be deemed to be associated with one or more workflow domain. As shown in Figure 7, organization `A` contains workflow domain `A` which contains two WfMSs `WA1` and `WA2`. Organization `B` contains two workflow domains, `B1` and `B2`, where domain `B1` contains WfMS `WB1` and domain `B2` contains WfMS `WB2`.

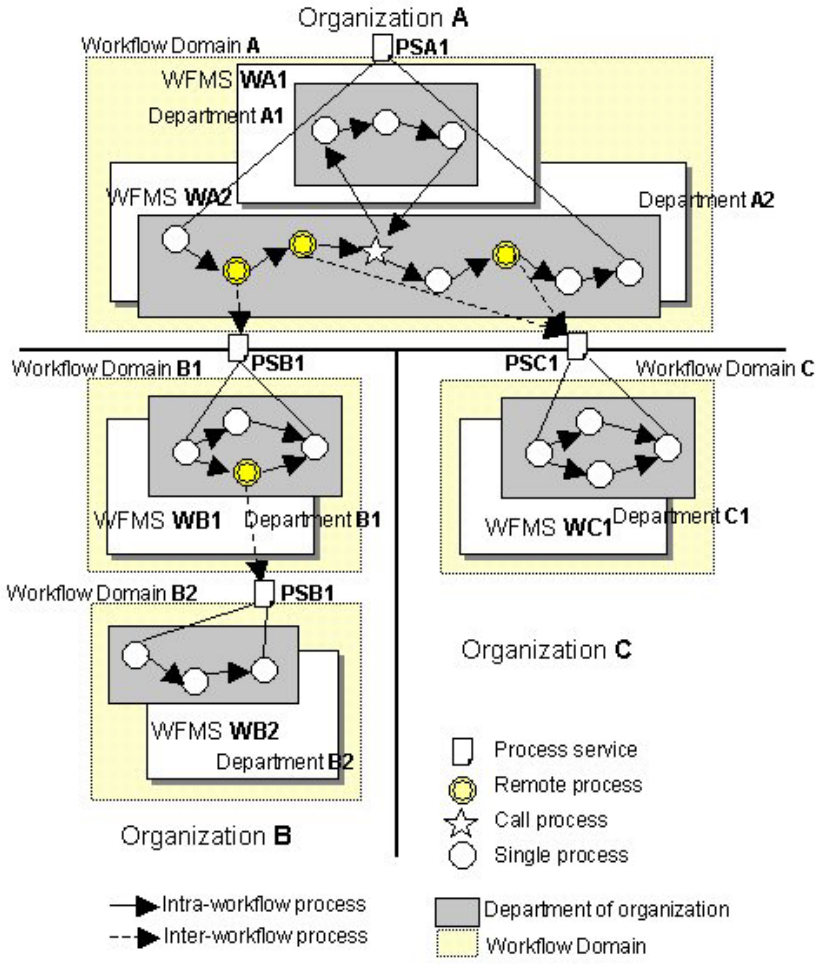


Figure 7. Call-Process Model (local/remote environment)

The activities of an inter-organizational process are provided by different organizations to achieve a common business goal. Each organization can design the corresponding activities within its organization into (intra-)processes and pack these processes as its Process Services. When a process P1 needs to integrate with a process P2 of another organization, P1 can use an activity that points to Remote Process RP_A to refer to the process service PS_A of the target organization for registration and thus P2 is enacted.

Each process service acts as a proxy for an intra-process and also defines the signature, the process arguments, and it returns the values of the intra-process. The inter-process uses a Remote Process (RP) to refer to a specified process service, and then the target process service (instance) is used to refer to the actual intra-process.

The workflow control and application data are passed through Process Call Data (PCD). The PCD is a mapping mechanism that specifies the communication data of two types: IN and OUT, between process service and remote process. When an inter-organizational workflow process runs to its RP step, a set of process data are passed to the corresponding process service for handlings. When the process service is completed, the result data are sent back to the RP for resuming the next activities. Based on parameter types, IN and OUT, a mapping mechanism is constructed between virtual parameters of a RP and physical parameters of a process service in our system, as shown in Figure 8. A Parallel-Event is used for RP to synchronize peer remote-process in whole inter-workflow process during cooperation.

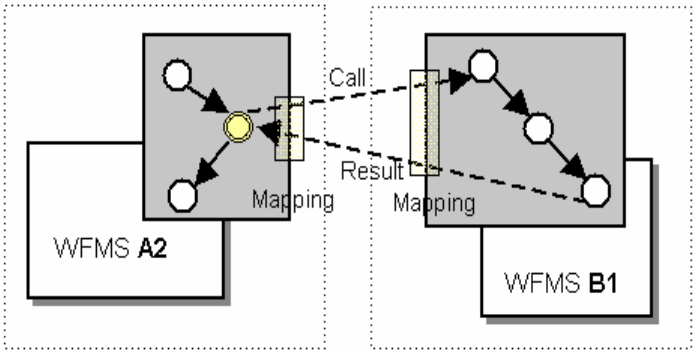


Figure 8. Mapping mechanism of parameters in an inter-organizational process

3.2 Integrated Workflow Component

In this sub-session, we present a cooperative workflow meta-model CA-PLAN to model inter-organizational processes. An inter-organizational process is a process whose activities are performed to cooperate, coordinate, and interact among different organizations over Internet. CA-PLAN extends the underlying model, PLAN [4], by adding remote processes, process services, and process control data as its modeling activity definitions, and by allowing a process in one WfMS to enact and monitor a remote process in another WfMS over network. In CA-PLAN, an inter-organizational process could be partitioned into several

parts based on organizations. CA-PLAN lets a workflow system inside an organization be modeled as an Integrated Workflow Component (IWC) as shown in Figure 9.

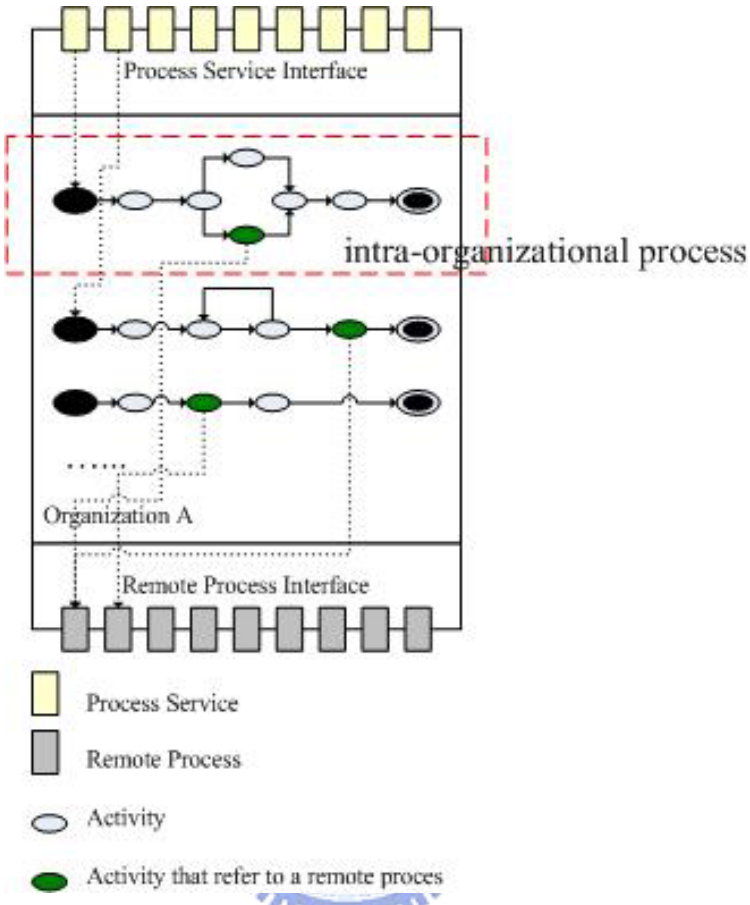


Figure 9. Integrated Workflow Component

Each IWC contains a process service interface specifying the process services provided by the organization, in conjunction with a remote process interface specifying the remote processes used in the intra-organization processes, and intra-organizational processes of the organization. Process service and remote process are input and output connector pin of an IWC. An inter-organizational process is defined as a routing path and assembled by IWCs through connection between their process services and remote processes. Each organization provides a set of process services for its partners. Each process service is referred to an intra-organizational process of the organization. An organization can integrate partner's process services by using remote processes. The activities/processes (of an intra-organizational

process) refer to the remote processes and then the remote process refers to a process service provided by partners.

For example, organization A in Figure 10 contains an intra-organizational process ProcessA1 which has an activity Act4 that refers to a remote process (RPA1). The RPA1 refers to Process service PSB1 of organization B. The PSB1 then refers to intra-organizational process ProcessB1 of organization B. The organizations A and B construct an inter-organizational process that links ProcessA1 and ProcessB1 by RPA1 and PSB1.

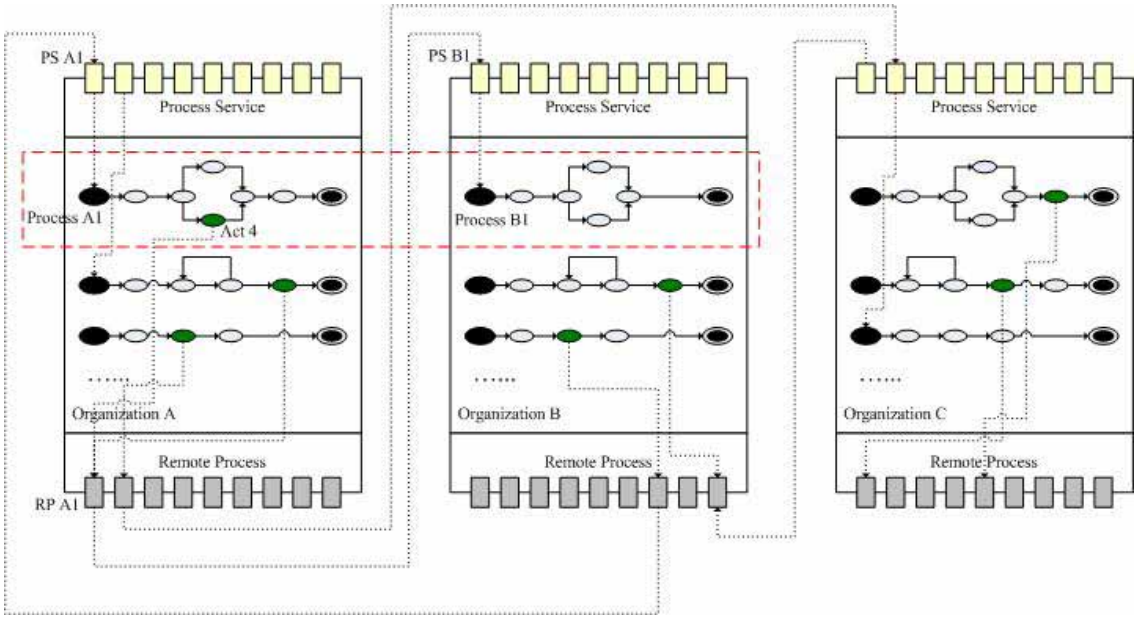


Figure 10. An inter-organizational process diagram

By using the indirect mapping mechanism for these two layers, an inter-organizational process can be partitioned into intra-organizational processes within different organizations. Whenever the corresponding process service has no change, an intra-organizational process may change to adapt the business requirements. This mechanism increase modularity, flexibility and reusability in design inter-organizational process. Besides, an intra-organizational process can be deemed as an inter-organizational process again, and this mechanism can be applied in a top down manner recursively.

There are three layers adopted for presentation and detailed handlings in CA-PLAN. The top layer lets designer devise an inter-organizational process as an intra-organizational

process (i.e., processes in WfMS). The medium layer, based on Java machine, provides a communication and management mechanism to help the design implicitly. This cooperative mechanism between WfMSs in CA-PLAN is modeled as a Remote Call Process (RCP) mechanism. RCP is a process-specific middleware that allows a process to enact a remote process as if it were a local (intra-organizational) process. RCP also supports the ability to manage, control and monitor the execution of an inter-organizational process instance. RCP hides the underlying mechanism of enacting a remote process, transporting process arguments and returns the values across the WfMSs. The bottom layer is a transport layer that may use http, web service, java rmi …etc.

In the top layer, the designer applies the tools in the workflow management system, (here we use Agentflow system to support the behavior) to write his/her inter-organizational workflow (processes). What he/she needs to concern is the existence of service process instances in corresponding sites and their input/output data. The tools in Agentflow system contain convenient graphical process design to pack a process service and to model inter-organizational processes. Agentflow also provides a process runtime environment to enact, control, and monitor the execution of inter-organizational processes.

On the other hand, the facility for following a workflow is an important tool. When a role finishes the work inside a workflow, i.e., when completing the interaction with a process inside the flow, he/she might want to trace the current status of the workflow for other work. The status information includes the current side(s) or handling role, and when is the work to be completed, etc. The tool providing the facility might allow all the participants who have completed his work to own the function. For those who do not get their job inside the workflow, they do not even know the existence of such a workflow. Therefore, they do not own the function. In a workflow system of single workflow engine, the tool can be built associated with the engine directly by constructing a flow map which can be read by the participants who own the functionality only.

In an inter-organizational workflow, a participant might not own such a function described as above. For example, the role inside the process enacted through process service is better to be hidden from the information outside the organization. It is a better design for the function to be owned by the participants in the organization where the inter-organizational workflow is enacted. On the other hand, for the security and information hiding, the status replied by a workflow of the organization different from the original is limited according to the organizational policy. The function and status information is different from that is traced inside the workflow system of an organization. There are two rules summarized: 1) The participants have the above-mentioned tracing function on the intra-organizational process (workflow) inside their organization, and 2) An inter-organization process (workflow) only allows the participants who are in the original organization to have the power. Besides, the information returned from an organization is decided by its policy respectively.

Obviously, the tool for the status tracing in an inter-organizational process is different from the one inside an intra-organizational process. The design and implementation of the tool will be discussed later.

The cooperative mechanism between WfMSs in CA-PLAN is named as Remote Call Process (RCP) mechanism. RCP is a process-specific middleware that allows a process to enact an external process remotely as local process. RCP also provides management, control and monitor of execution of an inter-organizational process instance. RCP hides the underlying mechanism of enacting remote processes, transporting process arguments and it returns values across the WfMSs.

RCP uses two process elements, Remote Process (RP) and Process Service (PS) to implement inter-process interaction across different WfMSs. The intra-processes are registered to a RCP-Registry and generate the corresponding PS. Each PS acts as a proxy to an intra-process and also defines the signature (what process arguments and return values) of the intra-process. The inter-process uses a RP to refer to a specified PS and then each PS

refers to the actual intra-process. Each PS is an integration process module that can be reused by the process in different organization through a RP. The mechanism allowing dynamic changes and reconfiguration can adapt dynamic and competitive business environment.

When a process designer wants to design an inter-organizational process, he/she can define the activities running on his/her organization, and integrate remote process services that pack the remote activities/processes provided by corresponding organizations through Remote Process. Through RCP, each organization can provide a set of process services for other organizations and use RP to integrate process services provided by other organizations.

A PS is a mechanism that packs a set of intra-processes for service accessed by the processes on other WfMSs, and it also specifies the associated arguments in and out. Each organization can provide multiple process services, each of which is a reusable process module that can be integrated within many inter-processes through RP. Furthermore, an inter-organizational process may integrate multiple process services provided by different organizations. For example, as shown in Figure 7, workflow domain B contains a process service PSB1 and the cooperative process packed as PSA1 in workflow domain A uses two process services: PSB1 from workflow domain B1 and PSC1 from workflow domain C.

As shown in Figure 11, a workflow domain may contain many processes enacted and controlled by the WfMSs in the domain. If a process in workflow domain A wants to become a service for another workflow domain, e.g., Workflow Domain B, it needs to be packed as a process service and registered to Process Service Repository (in Workflow Domain A).

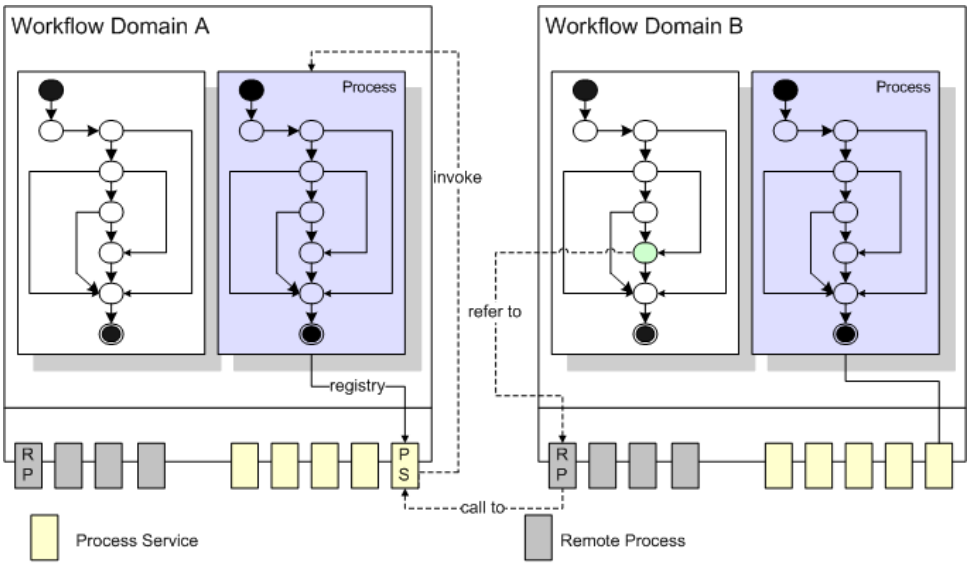


Figure 11. Process Service and Remote Process

A process service has two necessary actions: Pre-action and Post-action. Before a service process instance is enacted, the Pre-action is executed to initiate the data for the control return. At the end of a process instance, the Post-action is called to return control to the calling Process. A process service model element contains eight main attributes to specify a process service as shown in Table 1.

Table 1. The attributes of process services

Attribute	Descript
ProcessServiceId	The unique id of a process service
OrganizationName	The organization provides the process service
ProcessServiceName	The name of the process service
ProcessServiceDescription	The description of the process service
IntraProcessId	The wrapped process id of the process service. When the process service is enacted, the specified intra-process is enacted.
AccessControlList	Specifies the organization which can use the process service
IsMonitorSupport	Specifies whether the process service provides the monitor functionality and what information included.
MonitorLevel	If IsMonitorSupport is 'true' then PublicMonitor: process invoker can query all process service details including performer, progress and state of the process services PrivateMonitor: process invoker can query only state of the process services.
In parameter	
Out parameter	

3.3 Process Tracing and Security

Process tracing is an important aspect for workflow. Process monitor shows the history execution log and current status of an enacting process instance. The logs contain who and when the tasks were performed and the workflow-application data to be dealt with. The access control for process monitor is a security issue for workflow. Who can view the status of a process instance and what kind of the detail level of an executed task log is a security issue for different workflow application domain. Agentflow provides a default policy for process monitor. Each process participant, including process invoker and other process performers, can view the status of the process of the enacted process instance. Workflow-application executed data are more secure and sensitive data. CA-PLAN provides a role-base mechanism to control access control. The executed workflow application data logs can be viewed by its process performer and his superior only.

We define four monitor levels for process monitor as shown in Table 2. Level0 provides no monitor information and Level1 provides only current status and progress of a process instance. Level2 contains history execution log that shows the user and the time he/she performs the task. Level3 contains all Level2 data and every process instance's application data.

Table 2. Process Monitor Detail Level

Level	Description
Level0	No provides process monitor information.
Level1	Current status and progress of a process instance.
Level2	Includes Level1 data and history execution log data.
Level3	Includes Level1 and Level2 data and workflow application data.

Security issues between different WfMSs must be considered for inter-organizational Remote process execution and for monitoring across organizational boundaries. A process service contains AccessControlList attribute to specify the possible organization which uses the process service. IsMonitorSupport attribute specifies whether the process service provides

monitor functionality (traceable information). MonitorLevel attribute specifies the monitor level for the process service.

A Remote Process is a proxy mechanism that refers to a process service on another WfMS. When a cooperative process moves to a remote process node, the calling WfMS sends a request to the called WfMS whose corresponding process service will be enacted transparently. A remote process includes the following attributes: a workflow domain, an organization, a process service name, and arguments mapping. A remote process model element contains six main attributes to specify a remote process as show in Table 3

Table 3. The attributes of remote process

Attribute	Description
RemoteProcessId	The unique remote process id.
CalledOrganizationName	Specifies the organization name of the invoking process service
CalledProcessServiceId	Specifies the id of a remote process service
CalledProcessServiceName	Specifies the name of a remote process name
ExitCondition	Specifies the exit conditions of the remote process
EntranceCondition	Specifies the entrance conditions of the remote process

3.4 Cooperative Agentflow Process LANguage (CA-PLAN)

To describe the process definition from different perspectives, the Agentflow Process modeling LANguage(CAPLAN) is categorized into four sub-models(as shown in Figure 12), which include activity, artifact, role and project. The CA-PLAN is based on previous laboratory research [4][25].

These four models are:

- The *Activity Sub-Model* describes a workflow from functional & behavioral perspectives.
- The *Artifact Sub-Mode* describes a workflow from informational perspective of the workflow.
- The *Role Sub-Model* describes form organizational perspective of the workflow.
- The *Project Sub-Model* describes form project perspective of the workflow.

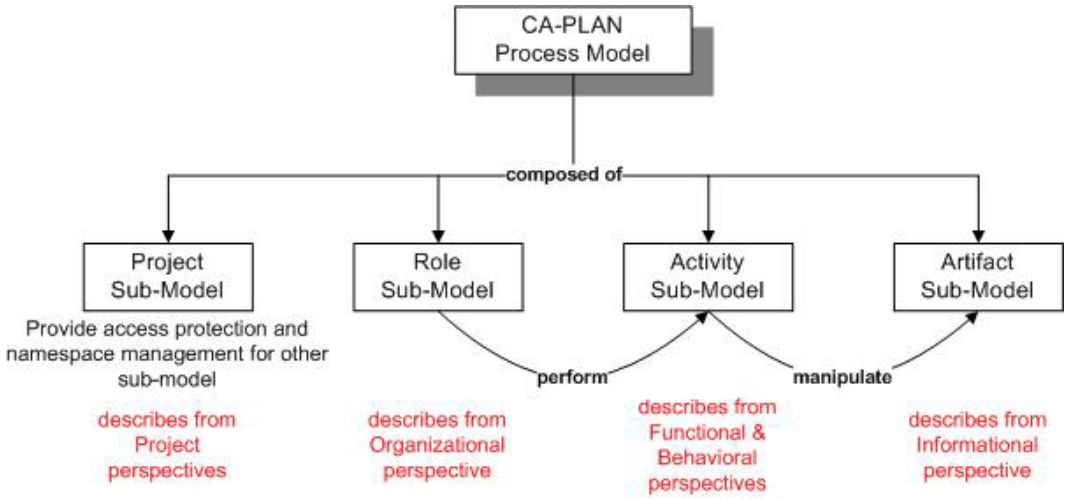


Figure 12. Cooperative Agentflow Process LANguage

The project sub-model provides access protection mechanism, namespace management, and project role model. To make processes easier to find and to use, to avoid naming conflicts, and to control access, programmers bundle groups of related processes and artifacts into projects.

The role sub-model, from an organizational perspective, depicts the process manager, the organization hierarchy and communication among the roles. Roles are abstractions of an organization model, which enable the representation of the procedure independently of the actual organization (how people are assigned to roles, how roles are grouped into organizational units). Using roles, and a database representing the organization of the enterprise, a workflow tool is able to assign automatically each activity (or work item) to the adequate employee inside the company.

The activity sub-model, from functional and behavioral perspectives, depicts the functionality of a process, its method of completion, and its initiation and termination. Also, it models various process behaviors required. The artifact sub-model, from an informational perspective, depicts reference materials and the lifecycle of an artifact. It captures two important relationships between artifacts, which are aggregation and dependency. The resource sub-model, from a project perspective, describes the type and number of required resources to ensure completion.

There are basic seven diagrams, four tree views, and fourth forms provided to assist a CA-PLAN modeler to conveniently describe their process, which are Enterprise Diagram, Project Diagram, Project Role Diagram, Process Diagram, Artifact Diagram, Artifact State Transition Diagram, and Organization Diagram. Table 4 lists the notations for a PLAN element, which are constructed or extended from above diagrams. For a software process, the diagrams form a bridge between guidance and enactment. Furthermore, they enable a process designer to model a software process. The graph/net-based approach is clear, comprehensible as well as suitable for communication, training and guidance. It allows process designers to model a software process in a higher-level abstraction.

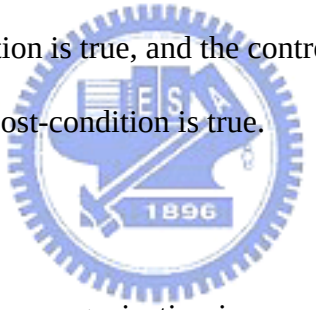
Table 4. Summarizes the notations of CA-PLAN

Sub-Module	Perspectives	Entities	Notations
Project	Project	Project	Enterprise Diagram (*ENT-Diagram*) Project Diagram (*PRJ-Diagram*) Project Definition Form (*{RJ-Form*) Project List (*PRJ-List*)
		Project Role	Project Role Tree (*PRJROL-Tree*) Project Role Diagram (*PRJROL-Diagram*) Project Role Definition Form (*PRJROL-Form*)
Activity	Functional	Activity	Process Tree (*PRO-Tree*) Process Definition Form (*PRO-Form*) Activity Definition Form (*ACT-From*) Called Process Definition Form (CALLPRO-Form) Countersign Form (*SIGN-Form*)
	Behavioral	Process Transitions	Process Diagram (*PRO-Diagram*)
Artifact	Informational	Artifact	Artifact Tree (*ART-Tree*) Artifact Diagram (*ART-Diagram*) Artifact State Transition Diagram (*AST-Diagram*) Artifact Definition Form (*ART-Form*) Artifact State Definition Form (*AST-Form*) Artifact Attribute Definition Form (*ATTR-Form*)
Role	Organizational	Process Participant, Company, Department, Member	Organization Tree (*ORG-Tree*) Organization Diagram (*ORG-Diagram*) Company Definition Form (*COM-Form*) Department Definition Form (*DEP-Form*) Role Definition Form (*ROL-Form*) Member Definition Form (*MEM-Form*)

A Process Diagram is a direct graph, where a node represents a set of activities and an arc between two nodes represents the flow from the end node to the head node. A node can be represented as a new diagram, if it contains a (business) process of more than one activity. A node which can not be decomposed again represents one activity handled by a role. There are some data passed through a flow.

An artifact state diagram is represented as a finite state machine. The artifacts represent the data between workflow nodes and worked by activities. In a workflow node associated with a role, an activity represents the work on artifact(s) by the role. For a node which can be decomposed into a new diagram, the activities will be described more details in the new diagrams.

There is a pre-condition and post-condition associated with a node, where a node is enacted only when its precondition is true, and the control is passed out only when the activity or process is executed and the post-condition is true.



3.5 Organization Model

In our workflow model, an organization is composed of a set of roles, where a person may represent one of more roles. Using roles, a workflow tools is able assign each activity to a proper employee automatically inside the company.

3.5.1 Organization Meta Model

There are two types of activities: automatic activity and human-intensive activity. For a human-intensive activity, human resource need be assigned to do the work. There are more than one organizations and hierarchies in an enterprise: departments, sub-departments, roles, employees, managers, etc. Organization view captures the human resource and the enterprise structure. Agentflow system uses an organization tree to describe an organization model of a company. The nodes of such a tree can be a department or a role. Leafs of the tree are

members. The hierarchical tree structure of an organization indicates the administrative relationship among the company, department and role.

Above hierarchical tree structure consists of four type nodes, Company, Department, Role, and MemberRecord as showed in Figure 13.

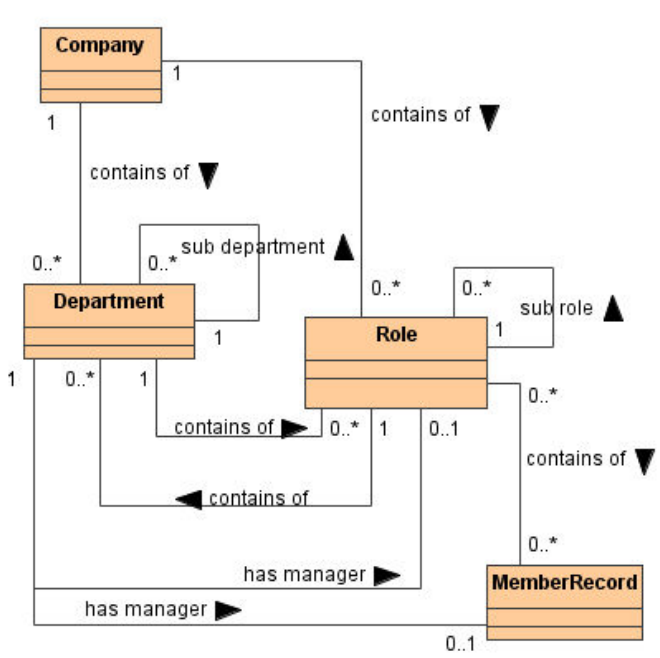


Figure 13. Organization Meta Model

A company indicating the root of an organization contains some departments and roles. A role could be assigned as the manager of the company. A department might contain sub-departments and roles. It may contain more than one role, of which one is the manager. A role might contain roles, departments and members. When a member plays a role in the organization, the member node is a child of the role node. A MemberRecord node indicates an employee of an organization. An employee could play one or more roles in an organization and it has only one main role.

3.5.2 Project Role Meta Model

A project is a collection of related processes, artifacts and project roles providing access protection and name space management. A project has a project manager and project

roles. In addition to formal organizational roles, there exist many casual roles due to some teamwork. We use project role to represent these situations. A project role is a group of employees. A project role could contain sub project roles. Figure 14 show the relationship between Project, Project Role, and MemberRecord.

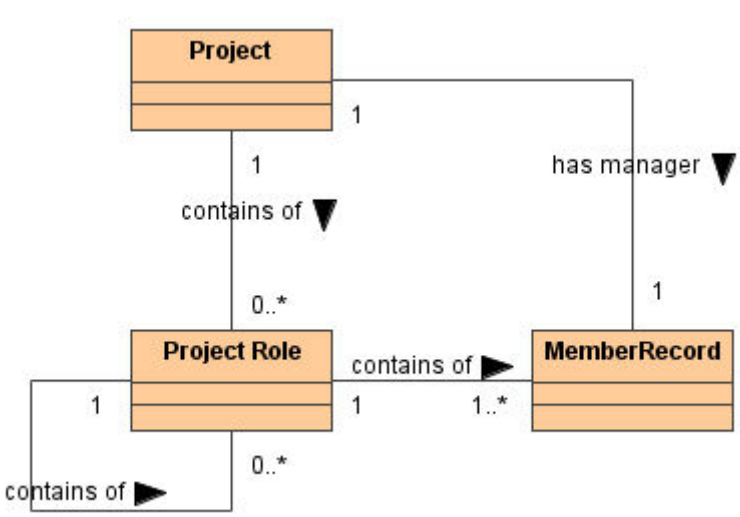
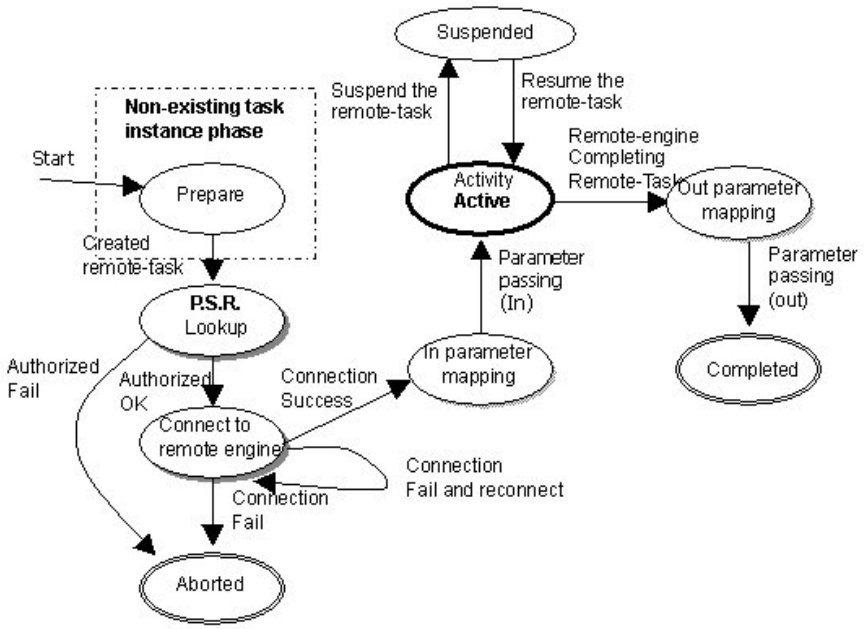


Figure 14. Project Role Meta Model

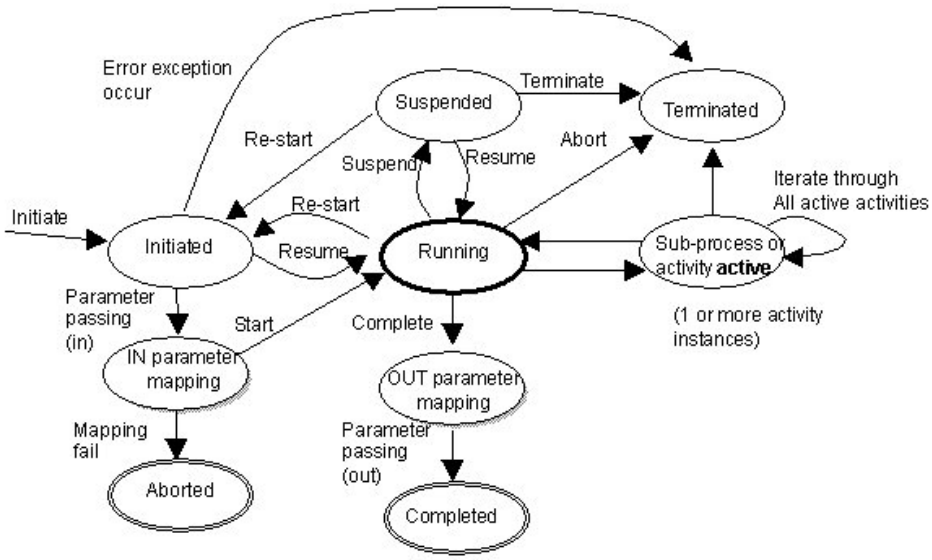
The hierarchical tree structure of a project role model indicates the administrative relationship between the project and project roles nodes. The root node of project role model is a Project node. The manager of a project administers its child project roles. A parent project role is the manager of the child project roles.

3.6 Remote Call-Process Life Cycle

A RCP is the cooperation mechanism of I-WfMSs of Agentflow, and its running styles can be classified into two operation modes, remote call-process calling (Remote Process in short) operation and remote call-process called (Process Service in short) operation, which serve enactment procedures separately within flow engines of different workflow domain. In this sub-session, these two operation modes are defined and discussed with finite state machines and both their life cycles are shown in Figure 15.



(a) Remote Call-Process Life Cycle of "**remote process**" engine's **(virtual) activity**



(b) Remote Call-Process Life Cycle of "**Process Service**" engine's **(physical)**

Figure 15. Remote Task Life Cycle

Figure 15 (a) indicates the state transition diagram of a runtime instance of remote process. When one activity is found as a “Remote Process”, the call-process calling I-WfMS (calling I-WfMS) creates a Remote Process instance and sets it to be “Prepare” state. According to the information of the Remote Process, the calling I-WfMS looks up the workflow domain and the Process Service and the instance enters “P.S.R. Lookup” state and then gets the authorization for looking up the remote called I-WfMS. When getting the

authorization, the instance enters “Connect to Remote engine” state to get real network address for connection. If the connection succeeds, the instance enters “In parameter passing” state and invokes the “In parameter passing” mechanism that forwards the parameters for remote Process Services. After completing all the operations, the calling I-WfMS forwards the parameters and delegates the flow control to remote called I-WfMSs for creating remote Process Service instance. Then the instance enters “Remote Activity Active” state until the remote Process Services instance completed the works. Finally, when the remote called I-WfMS completes the remote request, it sends the related OUT messages back to original calling I-WfMS_ and the instance enters “OUT message mapping” state. Then the calling I-WfMS completes the instance and triggers the next activity instance.

Figure 15 (b) indicates the state transition diagrams corresponding to the life cycle of a Process Service instance. The workflow engine of the I-WfMS (called I-WfMS) for a Process Service enters “initiated” state when such an instance is enacted. The corresponding instance enters “IN message mapping” state, when called I-WfMS receives “In parameter passing” operation message. Then, the instance enters “running” state; the called I-WfMS configures and launches the remote call-process instance specified by the Process Service. The Process Service may contain one or more activities, and those activities are enacted by called I-WfMS according to the process service definition, and the instance enters “subprocess or activity active” state.

In case the called I-WfMS is too busy, remote call-process is suspended, i.e., its state is set as “Suspended”. When called I-WfMS resumes this remote call-process processing, the state is backing to “running”. After the called I-WfMS completes all its operations, the instance enters “OUT message mapping” state for sending resulting message back to original calling I-WfMS. If the message passing is done, the called I-WfMS changes the instance’s state to be “Completed”, i.e., the process service completes and returns control to the calling I-WfMS.

4. The architecture and design of Agentflow system

In this chapter, we introduce the Agentflow system that implements the CA-PLAN and RCP mechanism to model, enact, control and monitor inter-organizational processes. Agentflow provides convenient graphical process design tools (PDE) to design intra- (and thus inter-) organizational workflow. Agentflow also provides a process runtime environment (PASE) to enact, control, and monitor the inter-organizational processes. Agentflow system is implemented in Java language.

Session 4.1 gives an overview of the Agentflow system. Session 4.2 describes the tools provided by Agentflow system. Session 4.3 presents the cooperative mechanism of Agentflow systems. Session 4.4 introduces collaboration mechanism between PDE servers. Session 4.5 introduces collaboration mechanism between PASE servers.

4.1 A overview of Agentflow

Agentflow System that was developed by our laboratory is a typical case of Workflow Management System. Agentflow System is n-tier software architecture. It contains three main parts as follow. Figure 16 illustrates the Agentflow System architecture [24].

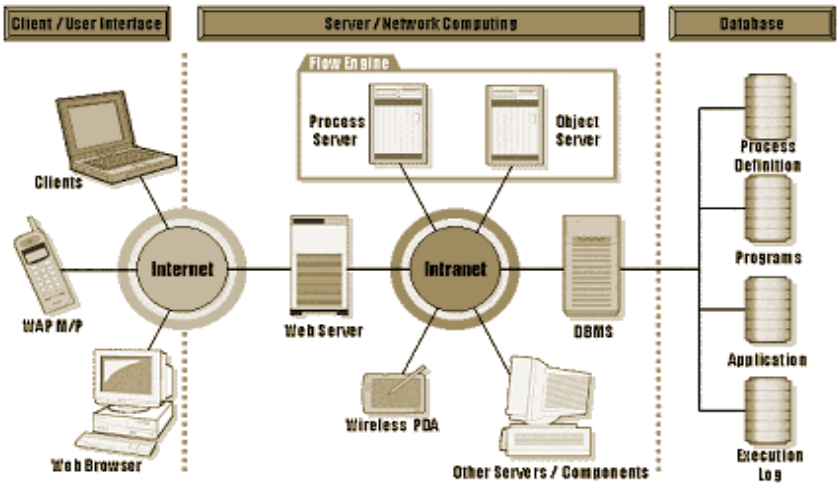


Figure 16. Agentflow System architecture

1: Client/User Interface

The main part of Client/User Interface is Agenda, which is a client side program of Agentflow System. Users can install Agenda on different platform and access Agentflow System through Internet. With Agenda, user can know the jobs that are currently needed to deal with the processes that can be initialized, and the explicit programs that need be called, and how to trace the processes. Users even do not require installing Agenda. They can just download Agenda through Web.

2: Workflow Server

Agentflow System utilizes the concept of network centric computation. In this concept, the network environment is thought of as a super computer and every program which user runs currently is directly downloaded to the client side. The workflow server plays the role of dispatching tasks to each user. And each client can receive the newest task list.

3: Database Server and DBMS

DBMS contains stored data, electric form, and Java program flow model and integrate other databases. Agentflow System is based on "A Process-Centered Software Engineering Environment with Network Centric Computing," PSEE [4], which provide modeling, monitoring, and measuring of software processes. Agentflow System provides two main software servers:

1: Process Definition Editor (PDE) Server

PDE Server provides a user/project access control mechanism and a Process Definition Repository that store process definitions. PDE provides a friendly user interface for process designers to define process models. Process designers can model a process with multiple viewpoints, including diagram, tree and form. With a viewpoint of diagram editing, process designers can depict executing sequence of processes and state-transition of artifacts. With a viewpoint of tree, process designers can easily sketch hierarchy of processes, roles, and artifacts. With a viewpoint of form

editing, process designers can define detailed information of each entity, such as synopsis, action description, informal specification, etc. For each process, designers can write scripts to describe some behavior within the process. These scripts, called action-scripts, can be interpreted by PASE Server to do some complex task automatically, for example, sending a mail to certain customer.

2: Process Aided Software engineering Environment (PASE) Server

Process models are enacted in the process enacting environments. This environment also provides functions such as process monitoring and execution records. PASE Server provides guidance, enforcement, automation, and different views of the process, tool invocations, and feedback for process improvements. All logs of workflow runtime data are stored in a Runtime Repository. PASE Server also provides an Agenda that shows the running processes and related information. End users use the Agenda to communicate with PASE Server, which controls and monitors the process execution, and cooperates with others. An important feature of the Agenda is the tool integration of existing developing tools, such as text editors, multi-media players and CASE tools. The Agenda automatically executes these tools defined in the process definition at modeling phase.

4.2 A Cooperative mechanism of Agentflow system

We constructed an Agentflow system by applying RCP mechanism on PDE and PASE servers, and by adding a name server called RCP-Registry. The process service makes an intra-process available for remote invocation by binding it to a name in the RCP-Registry. The remote process can thus check for the availability of a certain process service by looking up its name in the registry. The RCP-Registry thus acts as a central management point for RCP. The RCP-Registry is thus a simple name repository. It does not address the problem of invoking remote process actually.

RCP-Registry contains a Process Service Repository that stores the process service information of a workflow domain, such as a list of process service names, locations of PASE and PDE servers, organization name and organization telephone, etc. RCP-Registry does not send out any detail called process definition of a process service. For example, the executive roles of the processes are usually kept in secret.

Our framework follows two inter-organizational workflow rules: 1) a workflow designer may work on a unique portion of an inter-organizational workflow. 2) Multiple workflow engines, running in several distinct workflow domains and to be managed by their organizations, may work together to accomplish a mission. An example of Agentflow system is shown in Figure 17, where each organization (organization A and B) contains PASE server, PDE server, and database. Process designers are responsible for the design of flows with PDE that interacts with PDE server in their organization. Users in Agenda are responsible for communicating with PASE server and executing the flows that are designed by process designers. The database contains process definition, runtime instance, and organization repositories. RCP-Registry that contains a Process Service Repository is used to register and lookup the process services of a workflow domain.

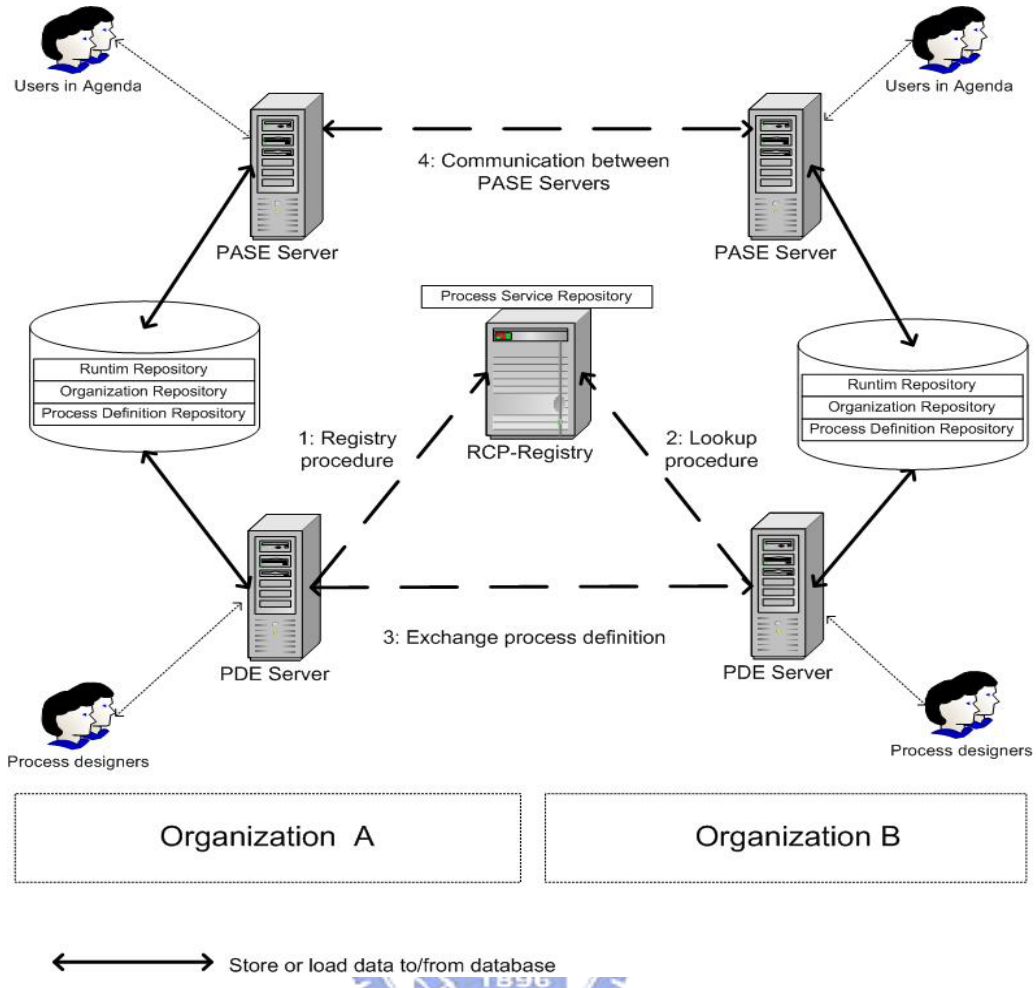


Figure 17. A cooperative framework of Agentflow system

4.3 Collaboration between PDE servers

4.3.1 PDE Server Architecture

The architecture of Process Definition Editor (PDE) is shown in Figure 18. The PDE Server contains a Project/User Access Control module to control access mechanism. The ObjectBroker converts between objects and relational database module. It converts objects to database table records and database records to objects. DB Adapter is a database adapter to adapt specific database server. The PDE Server operates the Organization, Process Definition and Process Runtime repositories. The communication mechanism between PDE Server and PDE is java RMI mechanism. The PDE is designed by Model-View-Control Model. The PDE contains four data model - PDE, Organization, Project and Application Model.

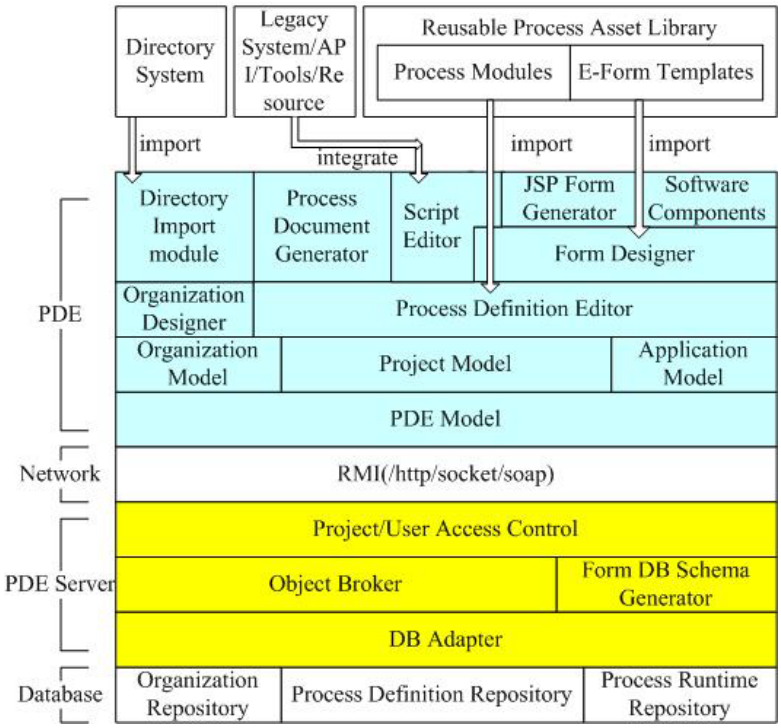


Figure 18. Architecture of process definition editor

4.3.2 Interoperation between PDE Servers

PDE and PDE Server are used to model workflow processes in an organization. To achieve workflow processes that communicate between different WfMSs, again, we develop the techniques with process service and remote process in PDE Server.

To create an inter-Organizational workflow executed in different organizations, there are three steps executed between PDE Servers: register procedure, lookup procedure, and exchange process definition procedure.

The first service of RCP-Registry is to provide a process service registration mechanism to PDE servers. If a process designer wants to provide a process for other workflow domains, i.e., to be called by the process in other external workflow domains, he/she firstly needs to pack and register the process as a process service on RCP-Registry. Figure 17 introduces the collaboration between PDE servers. When process designers in organization A want to provide service processes used by organization B, they will use PDE server to connect RCP-Registry and send a register procedure containing service related

information to RCP-Registry. When a process designer in organization B wants to make up a cooperative process, he/she needs to use PDE server to connect RCP-Registry and to send a lookup procedure to RCP-Registry. The lookup procedure retrieves the information of the register procedure so that process designers can get the organization information of the services. Figure 19 shows the information of lookup procedure

Because register and lookup procedures provide and receive the organization information only, they do not exchange process definitions. If an organization registers process definition in RCP-Registry, the process definition registered may change later. After selecting the available processes, PDE server in the original organization will automatically send a Remote Method Invocation (RMI) [26] request to the PDE server in the target organization to exchange process definition. A graphical tool is designed to show the process definitions to be exchanged.

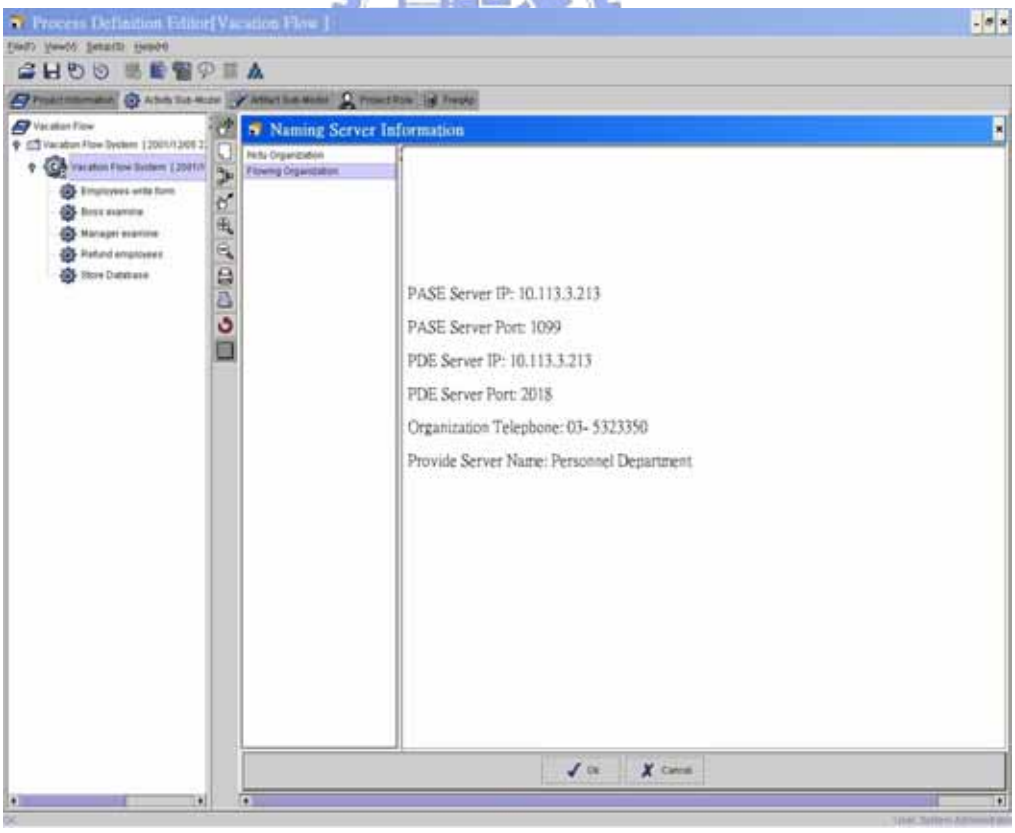


Figure 19. The information of lookup procedure

The automation tool provides three functions below:

1. It can provide available processes, including process view, artifact view, and sub-processes view, to other organizations that already own workflows with their PDE servers.
2. The edit tool allows process designers to model complex and various processes, some of which come from other organizations, into their workflow processes and to create program modules that are needed in runtime phase. But, they do not have the authority to modify the available processes. They can decide the Start/Complete conditions and provide the corresponding artifacts of the available processes.
3. The edit tool also allows user to design the Process Call Data of a Process Service and assign (IN and OUT) parameters mapping relationship between a Remote Process and a Process Service.

4.4 Collaboration between PASE Servers

4.4.1 PASE Server Architecture

Agentflow runtime enactment system, called PASE server, drives the flow of work and manages the workflow database. PASE server also facilitates process enacting, control, management, and monitoring. Figure 20 shows the PASE server architecture. Server service provides a common application programming interface for other workflow clients, engines, or administration.

`ServerService` plays the role of a service interface between workflow clients (e.g. end-users or other WfMSs) and workflow kernel service that provides the ability to create, query, and dispatch task instance…etc. `Organization Manager` provides functions to access organizational elements (that include companies, departments, projects, roles, and members) and to calculate the relationship between them. `Transaction Manager` is responsible for managing transaction mechanism in workflow engine. `ObjectManager`

manages an object cache for reducing the communication overhead between PASE server and database. `ObjectBroker` maps the objects to database. `DBAdapter` provides a high level database function call that wraps JDBC API to store and access database storage.

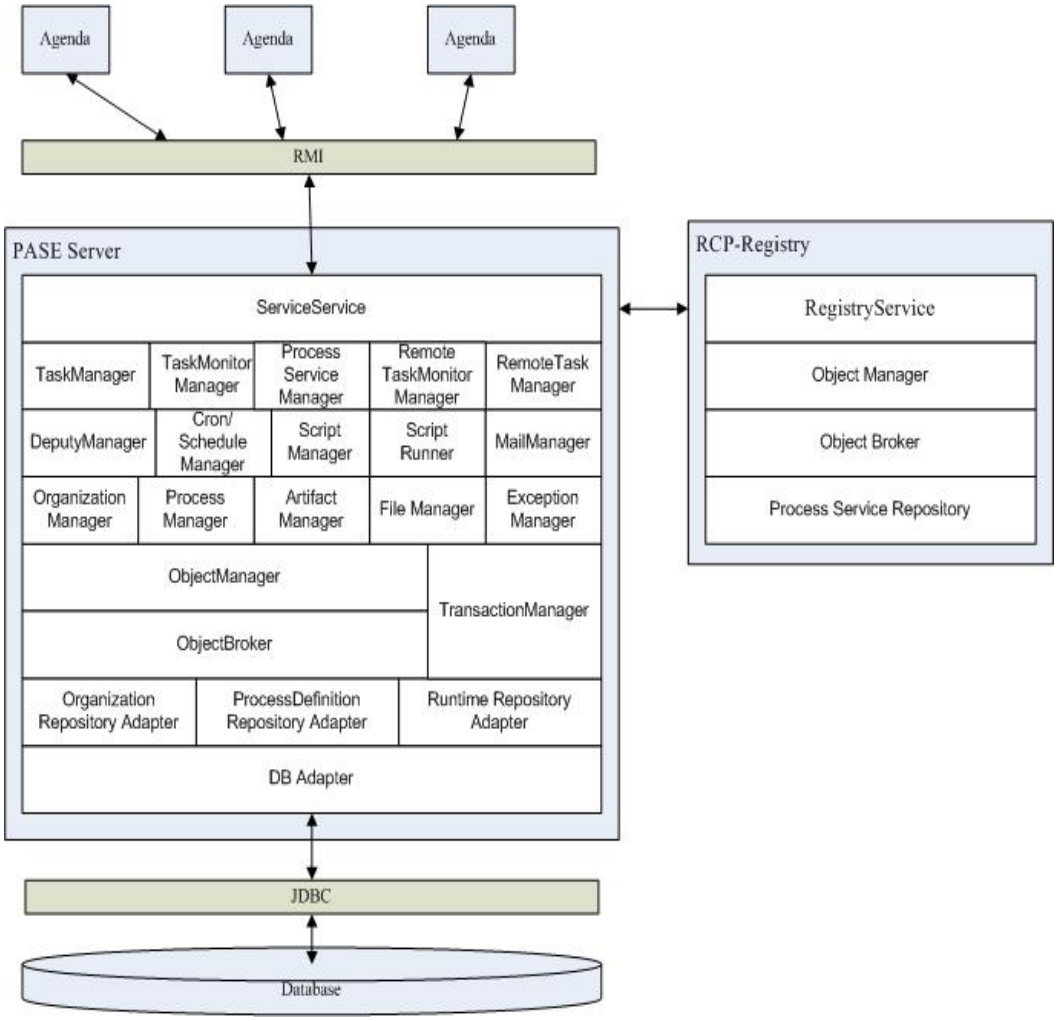


Figure 20. PASE server architecture

When a new process instance is initiated, the workflow engine also creates a corresponding `TaskManager(TM)` for managing the whole process execution and run time information like state, parameters… etc. `TaskManager` manages controls, and routes tasks. `RemoteTaskManager` manages remote tasks on other WfMSs. Each `TaskManager` has a `TaskMonitor` that contains process definition, including conditions and flow paths, and controlling the process execution. The `ArtifactManager` deals with artifact instance information. When `ServerService` receives the request of remote process call from

another WfMS, it passes this command to `TaskManager`. After `TaskManager` initiates the remote call-process instance, it also creates a `RemoteTaskManager` for managing the remote call-process instances. `RemoteTaskManager` contains a `RemoteTask Monitor`, a remote process proxy for remote call-process execution, to be used in tracking and responding events from remote WfMSs.

4.4.2 Interoperation between PASE Servers

This sub-session will discuss the interoperation between PASE servers. The communication example between PASE servers is shown in Figure 21. Consider a cooperative process that is made up by PDE servers in two organizations A and B. When user C triggers a cooperative process with Agenda, the latter notifies PASE server to 1) retrieve the cooperative process model from database by using object cache and management, and 2) enact and control the process of the cooperative process. Once a cooperative process is enacted in step 2), PASE server shall create a TM that manages all tasks generated by the cooperative process in runtime. The TM is also responsible for notifying user(s) to do their task. This work is completed after the TM checks the organizations of the TM's tasks. The tasks in local organization (A) will be sent to corresponding users (D, E) by TM automatically. Otherwise, TM will search the RCP-Registry, find the location of PASE server in corresponding organization (B), create a RTM and send a RMI request to target PASE server. Finally, this PASE server creates a TM to notify the users in organization B, such as users F, G, and H, to complete the request task.

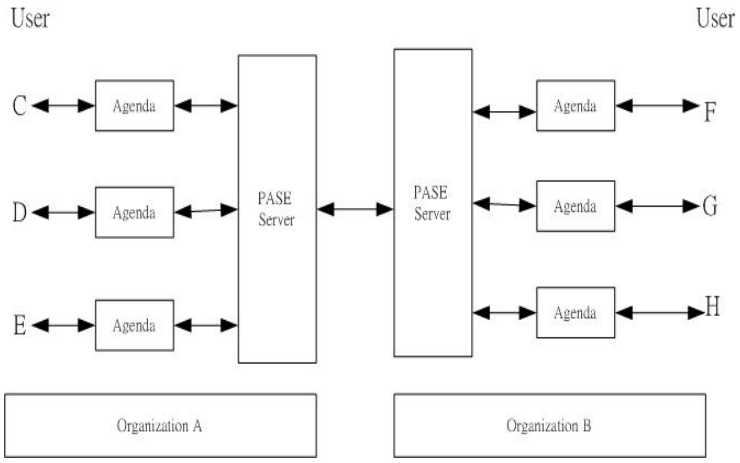


Figure 21. The communication example between PASE servers

Note that when a request task is complete, TM in organization B might send a message to notify the PASE server in organization A. The resulting message includes artifact states, process state, complete-condition, etc, and the PASE server follows with the message.

4.4.3 Remote Monitor

Process monitor is an important functionality for workflow engine. Process monitor can be used to track and show the status and progress of a process instances. By using process monitor, workflow participants can know the current status, progress, the activity step in a workflow, and the participants who are in charge of an enacted process instance. Agentflow provides a graphical process monitor for workflow participants to track and monitor the status and progress of a process instance. Through process monitor, user can know the above-mentioned process information for the process he enacted. User also can know status of a process instance that he has handled.

A key design issue is to monitor each cooperative process between organizations. In PASE server, each TM will create a `TaskMonitor` that monitors task states. Assume that user C in organization A wants to monitor a cooperative process which executes in organization B, he/she will notify the PASE server to find the `TaskMonitor` which manages the cooperative process. When the `TaskMonitor` detects the monitor process that

belongs to organization B, a `RemoteTaskMonitor` will be created and the PASE server will be notified to find the location of PASE server in organization B by Objects Cache and Object Management. After finding the location of organization B, local PASE server will send a request message to the PASE server in organization B. When the PASE server in organization B receives the request, the corresponding `TaskMonitor` in organization B will be found. The latter `TaskMonitor` starts to search the monitor structure for the cooperative process and it returns these monitor data to its PASE server.

5. Implementation of Agentflow

Agentflow System is implemented in Java, which is an object-oriented language. It helps program developers to reduce the development time and improve system performance. Agentflow System provides some types of database to communicate, such as Microsoft SQL Server, MySQL, Oracle, SyBase, and Informix.

5.1 Task Manager Algorithm

In this session, we discuss the task handling algorithm in Agentflow system as shown in Figure 22. When user enacts a process in Agentflow system, the system first checks whether the request is local. Then the associated TaskManager and TaskMonitor are constructed to handle the workflow instance. If the process is enacted by other WFMS, i.e., a remote invoked process is created a generated ticket is returned to the calling WFMS. Now a new task is created and run. There are four types of tasks: *Root Task*, *Compound Task*, *Single Task* and *Remote Task*. A Root Task represents a workflow instance. A Compound Task contains child tasks. A Single Task has no child task. A Remote Task is an instance of a Remote Process. The Root or Compound Task is expanded to create child task during execution. Furthermore, the workflow engine checks the task is a human-task or an automated-task. If the task is an automated-task, then the workflow engine executes the corresponding script. If the task is a human-task, workflow engine binds the artifact instance to the task and dispatches it to the executor located. User uses workflow client application (Agenda or Enterprise Process Portal) to operate or monitor the assigned tasks. When user completes a task, the result is returned to the workflow engine. When the task completes, the workflow engine checks whether there is next task. If yes, workflow engine creates next task for execution. The execution continues until no task exists.

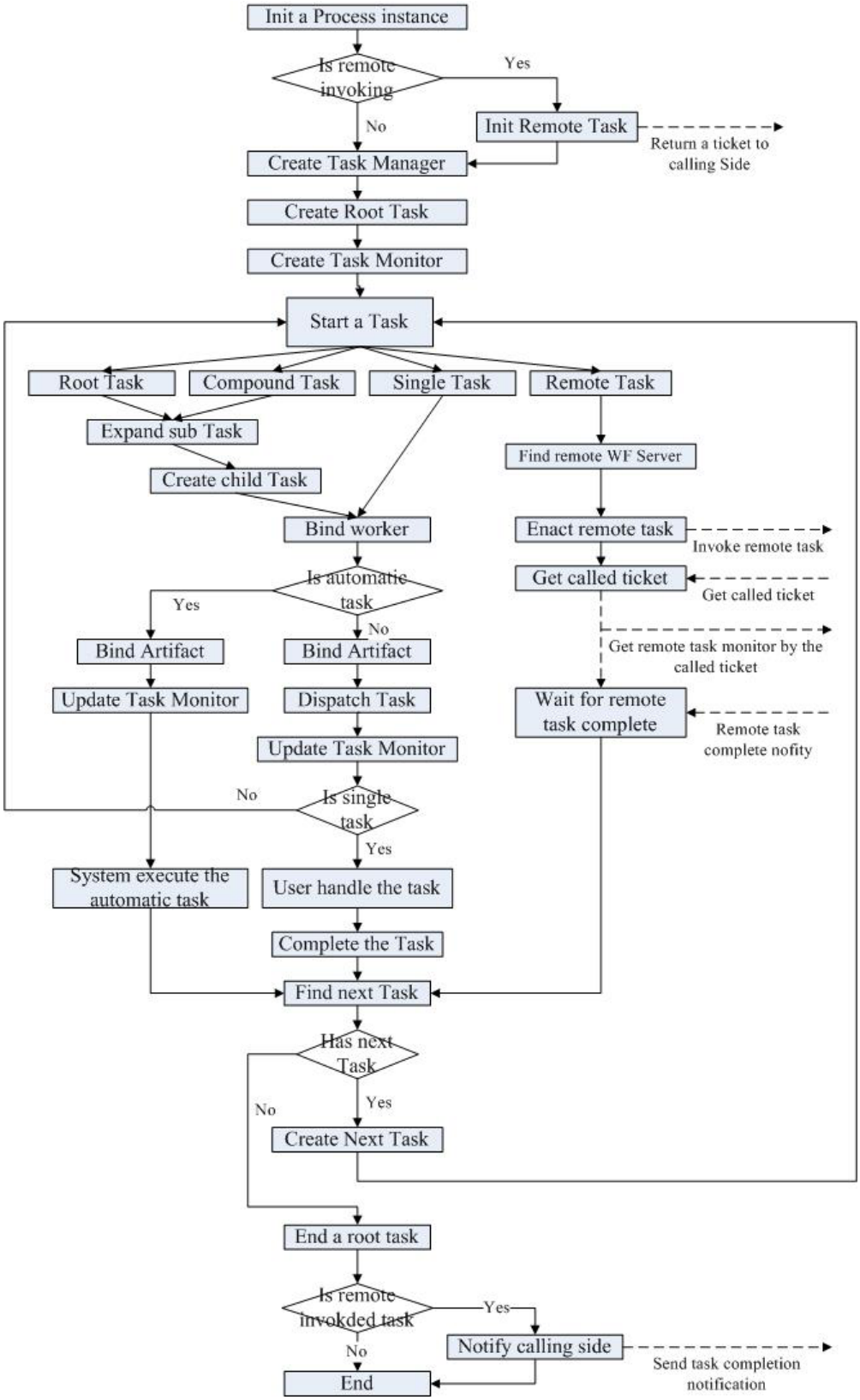


Figure 22. Task Manager Algorithm

In case the task is a remote task, workflow engine finds the location of called WFMS firstly. Secondly, the workflow engine notifies the remote WFMS that enacts the remote process. Then, the workflow engine waits to receive the called ticket. After the ticket is

received, the task keeps wait until the notice of the remote process. During the waiting time, the task can use the ticket to query the status of the invoked remote process instance.

5.2 Communication Mechanism between PDE Servers

To create a cooperative process executed by different organizations, there are three steps executed between PDE Servers, including registry procedure, lookup procedure, exchange process definition procedure, describes in this session.

The communication mechanism between Name Server and PDE Server contains registry procedure and lookup procedure. The objective of these two procedures is that process designers in PDE Server can know the workflow processes that can be used. The Java Naming and Directory Interface (JNDI) [27] are designed to simplify access to the directory infrastructure used in the development of advanced network applications. Traditionally, different APIs are need to access different directory services such as Lightweight Directory Access Protocol (LDAP) [28] or Sun's Network Information Service (NIS). However, JNDI supplies a standard API to access any type of directory services. The LDAP was developed in the early 1990s as a standard directory protocol. LDAP defines how clients should access data on the server and does not specify how the server should store the data. Most popular directory services now have an LDAP interface of some type, including NIS, Novell Directory Services (NDS), Windows NT Domains, Active Directory Services (ADS), etc. Therefore our PDE Server uses JNDI to connect the LDAP specification Name Server, such as Netscape iPlanet Directory Server.

5.2.1 Registry Procedure

Registry procedure provides available processes used by other organizations. Figure 23 shows the registry procedure. When process designers want to provide available processes, `Request_Registry()`, called by process designers, notifies PDE Server to register by

RMI mechanism. PDE Server receives the notice and calls `Connect_NameServer()` to connect Name Server by JNDI mechanism. `Connect_NameServer()` will return connection result. If connection succeeds, PDE Server will call `Registry_to_NameServer()` to register organization information, such as organization names, organization telephones, service names, and the locations of PDE and PASE Servers, etc. If connection fails, PDE Server will notify process designers the failure message of registry procedure.

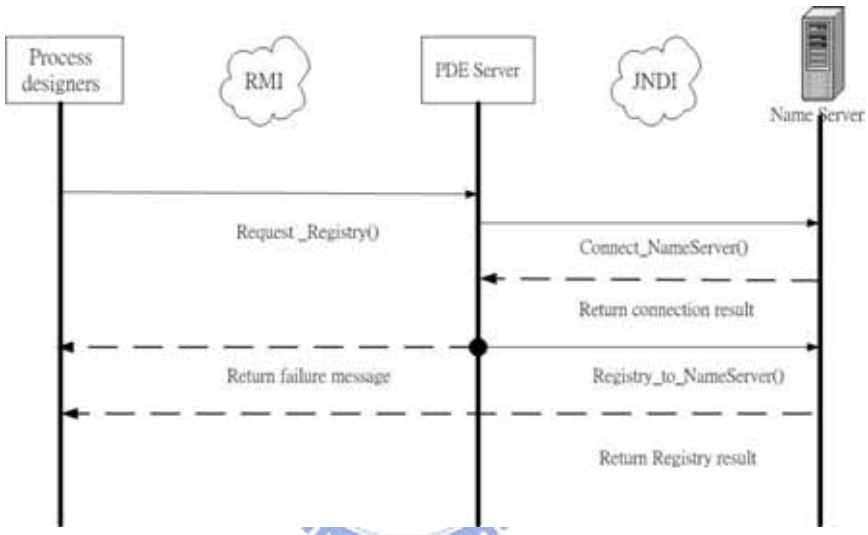


Figure 23. Registry procedure

5.2.2 Lookup Procedure

Lookup procedure searches the services that can be used. Figure 24 shows the lookup procedure. When process designers want to know the services which can be used, `Request_lookup()`, called by process designers, notifies PDE Server to look up by RMI mechanism. PDE Server receives the notice and calls `Connect_NameServer()` to connect Name Server by JNDI mechanism. `Connect_NameServer()` will return connection result. If connection succeeds, PDE Server will call `lookup_to_NameServer()` to look up organization information created by registry procedure. If connection fails, PDE Server will notify process designers the failure message of lookup procedure.

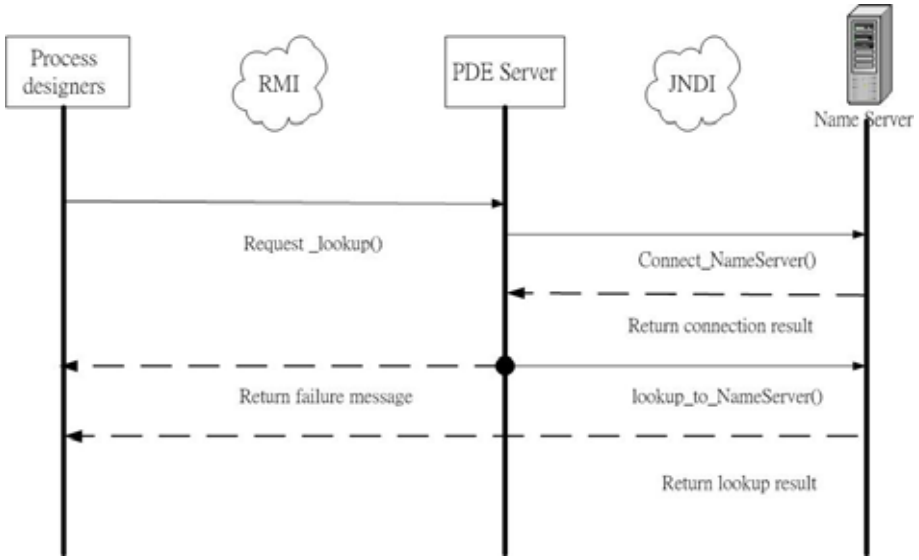


Figure 24. Lookup procedure

5.2.3 Exchange Process Definition Procedure

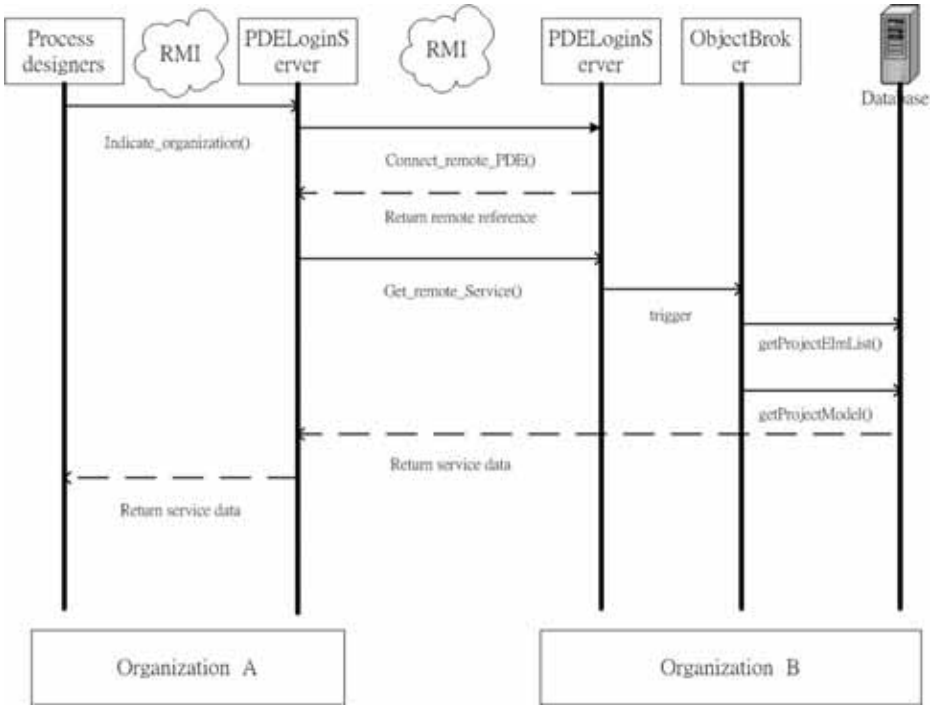


Figure 25. Exchange process definition procedure

When process designers in PDE Server complete the lookup procedure, the next procedure is exchanging process definitions between PDE Servers. Figure 25 shows the exchange process definition procedure. Because our PDE Server is implemented with object-oriented concept, we design the interface "Login", which originally takes as communication interface between process designers and PDE Server. Additionally some methods are added to

take as the communication interface between PDE Servers. In Figure 25, PDELoginServer is an object that implements the interface "Login". ObjectBroker is an object which managers database.

In organization A, process designers select available processes provided by organization B, `indicate_organization()`, called by process designers, will notify PDELoginServer to exchange process definitions. PDELoginServer receives the information of `indicate_organization()` and know the location of organization B. PDELoginServer will call `Connect_remote_PDE()` to connect PDE Server in organization B. `Connect_remote_PDE()` will return remote reference of PDELoginServer in organization B. When PDELoginServer receives the remote reference, it will call `Get_remote_Service()` to obtain process definitions which come from organization B. After PDELoginServer in organization B receives the notification of `Get_remote_Service()`, it will trigger ObjectBroker in organization B and ObjectBroker in organization B will call `getProjectElmList()` and `getProjectModel()` to load service data, including Process data, Artifact states, service names, project Diagrams, etc, from database in organization B. When ObjectBroker in organization B loads the data completely, it will notify PDELoginServer in organization B to return the service data to PDELoginServer. Last, PDELoginServer will send the service data to process designers.

When process designers in organization A receive the process definitions which come from organization B, they start to model a cooperative process executed betweens organizations A and B. PDE Server is responsible for storing process definitions into database. PASE Server is responsible for loading process definitions from database and executing process definitions in runtime. The cooperative processes have to be recorded completely, there are two additional tables used to record the information in organization B: remote_art_state and deliver map. The remote_art_state table records the state conditions of

article. PASE Server triggers tasks based on artifact states. When process designers design a cooperative process, they use artifact states in organization B to model a cooperative process. Besides recording the artifact instance information, the locations of PASE and PDE Serves are also recorded. Because PASE Server deals with the cooperative process in runtime, the organizations which have to execute the cooperative process are necessary. The deliver_map table records the relative locations of PDE and PASE Servers.

5.3 Communication Mechanism between PASE Servers

Figure 26 shows the communication procedure between PASE Servers. Figure 26, RMI_Server is an object that manages all workflow procedures. TM is an object that manages tasks created by workflow processes in runtime. ObjectBroker is an object that manages database information. In organization A, when user in Agenda triggers a cooperative process, he will send a notification to RMI_Server. RMI_Server will trigger a TM. TM loads the cooperative process from database and enacts the tasks for the cooperative process. After TM enacts the tasks, it will notify RMI_Server the users which are in Agenda and have to execute the task. RMI_Server will communicate with users in Agenda. When RMI_Server discovers the next task which belong to organization B, RMI_Server will search the deliver_map table and find the location of PASE Server in organization B. RMI_Server call `Connect_remote_PASE()` to connect PASE Server in organization B. `Connect_remote_PASE()` will return remote reference of PASE Server in organization B. After RMI_Server get remote reference of PASE Server in organization B, it will call `Registry_RMI_Server()` and `Start_cooperative_process()`. `Registry_RMI_Server()` is responsible for providing remote reference to PASE Server in organization B . `Start_cooperative_process()` is responsible for notifying PASE Server in organization B to execute the request task.

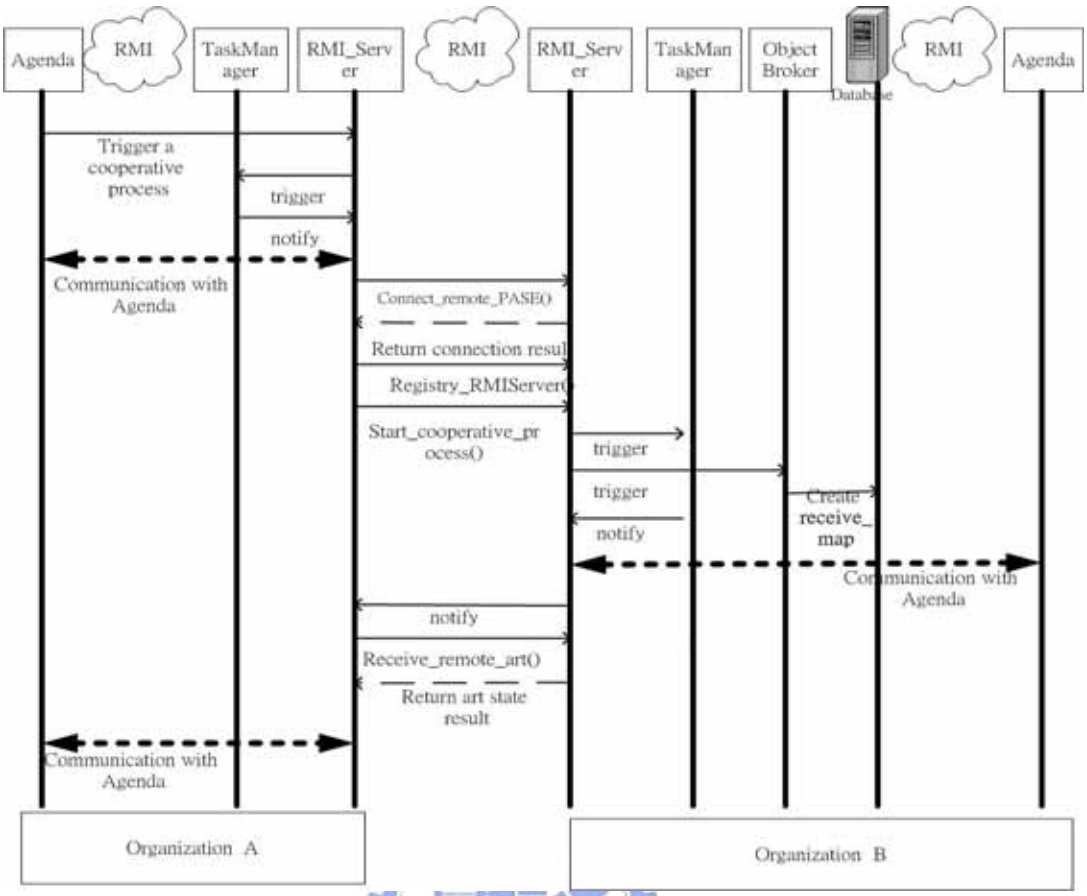


Figure 26. Communication procedure between PASE Servers

When RMI_Server in organization B receives the request message, it will trigger a TM to enact the request task and trigger ObjectBroker to create a database table, called receive_map table. The receive_map table records the location of PASE Server and the relative relationship of tasks between organizations A and B. After TM in organization B completely enacts the request task, it will notify RMI_Server in organization B to dispatch the request task to users in organization B.

After RMI_Server in organization B completely executes the request task, it will search receive_map table in database in organization B and find the location of PASE Server. RMI_Server in organization B connects to PASE Server and sends a notice message to RMI_Server. After RMI_Server receives notice message, RMI_Server will call `Receive_remote_art()` to receive the artifact state belong to the request task executed

in organization B. Last, RMI_Server decides the next task based on art state, which `Receive_remote_art()` receives, and dispatches the next task to users.

5.4 Synchronization Monitor between Organizations

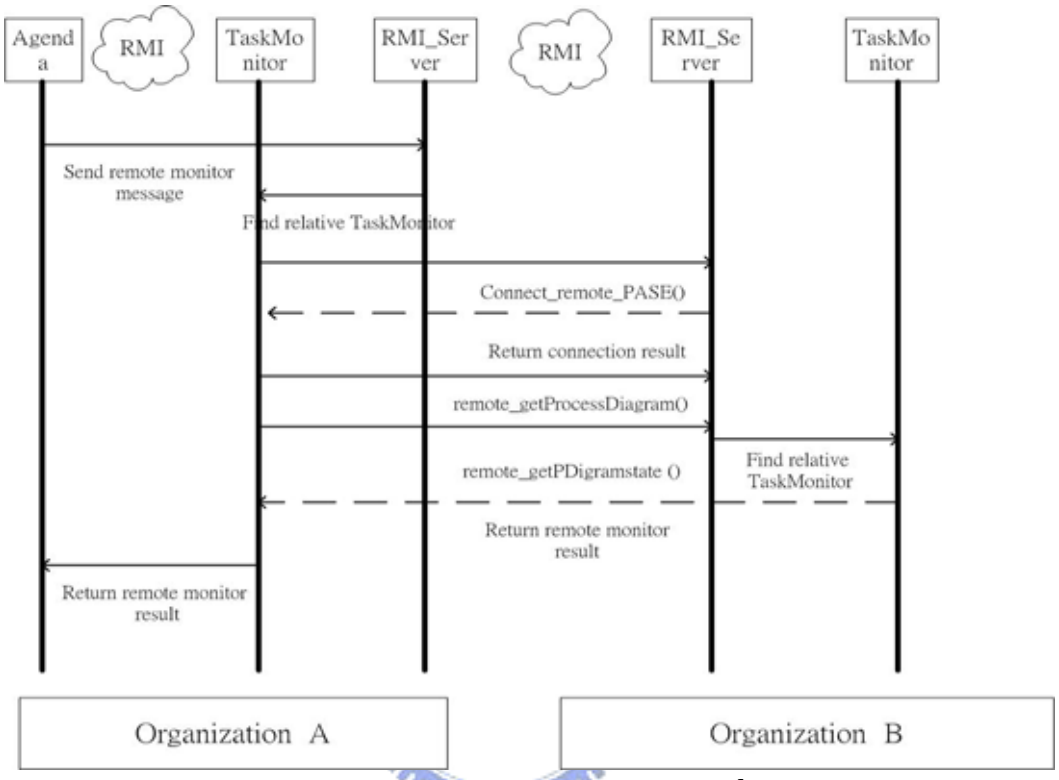


Figure 27. Remote monitor procedure

Figure 27 shows the remote monitor procedure. RMI_Server is an object that manages workflow procedure. TaskMonitor is an object that monitors task states. We will use Figure 13 to introduce the communication mechanism of remote monitor. Let's consider in organization A, when user wants to monitor a cooperative process, he will send the remote monitor message to RMI_Server to find the TaskMonitor which is responsible for monitoring the cooperative process. When TaskMonitor discovers the cooperative process executed by organization B, it will search the deliver_map table to find the location of PASE Server in organization B. TaskMonitor calls `Connect_remote_PASE()` to get remote reference of PASE Server in organization B. After TaskMonitor gets remote reference of PASE Server in organization B, it will call `remote_getProcessDiagram()` and `remote_getPDigramstate()` to notify PASE Server. The

`remote_getProcessDiagram()` is used to get the graphical model of cooperative process. The `remote_getPDigramstate()` is used to get the states of cooperative process. When RMI_Server in organization B gets the notification of `remote_getProcessDiagram()` and `remote_getPDigramstate()`, it will find the corresponding TaskMonitor in organization B. TaskMonitor in organization B finds the monitor state of cooperative process and returns the result to TaskMonitor back.

6. An Intelligent and Personalized Enterprise Process Portal

Workflow-based application systems are prevailing in the enterprise information environment. The scenarios of workflow application are no longer limited to the small user groups operating the document management or data processing applications. Nowadays the deployment to install workflow applications within the Enterprise Process Portal is emerging in the enterprises, and this helps the employee's assistants in daily (business) activities as well as manager's advisor in executive decision. Meanwhile, business process automation and integration is getting more sophisticated. Facing the varieties of user groups and application requirements, workflow systems need certain enhancements according to the following two reasons:

Firstly, user groups have different skills for information technologies. The diversity of workflow tools and front-end interfaces are essential for various user groups. Unitary front-end interface in conventional workflow systems cannot satisfy different professional demands.

Secondly, the application requirements of business process automation are getting sophisticated. Although the core workflow infrastructures still work for the applications, it may adopt the intelligence of software agents to empower the delivered systems and simplify the application operations, especially when the business process scenarios are better modeled as the interactions among the workflow system, human user and software agents.

Session 6.1 briefly introduces the software agents and demonstrates the typical scenarios of using an Enterprise Process Portal powered by an agent-based workflow engine. Session 6.2 explains what software agents help users by the intelligence inferred from pre-defined rules, as well as the resource, schedule, and member activity monitored by the workflow systems. Session 6.3 describes the software architecture for workflow-based enterprise portal techniques to support tool diversity and personalization in the workflow

workspace. It also discusses the interactions among end-users, workflow systems, and software agents.

6.1. Software Agents and Enterprise Process Portal

The research of software agents [5][6] during last decade has accumulated lots of outcome and demonstrated its applications in different areas. Software agents have several useful characteristics such as mobility, autonomy, and intelligence. They are also proactive to their executing environments. Therefore, researchers and software industry which work on workflow technologies, start to adopt software agents as part of the software design.

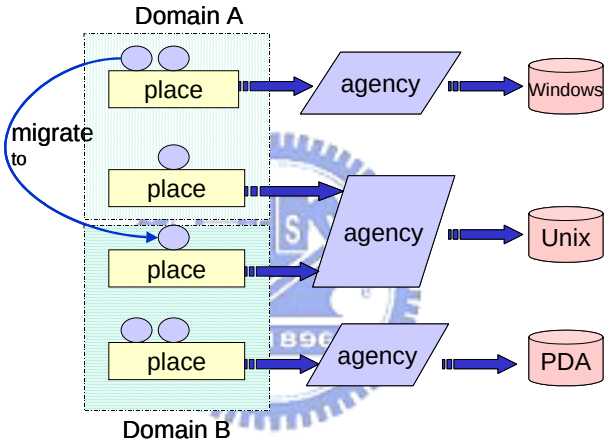


Figure 28. A typical software agent environment

Fundamental software agent researches mainly focuses on knowledge representation of agents, the ways that agents learn, make decision, communicate and negotiate so are the security and mobility are also issues of agent research. For software agents, '*places*' manage the executing context of agents by providing resources and setting policy. Agents migrate between places. Several places can be logically grouped into a *domain* (or called *region*). An *agent system* (or called *agency*) can be presented to make the differences of hardware or operating system transparent to *places*.

The mobility characteristic of software agents can help to describe workflow processes. For example, when the resources required to perform the activities of the process are placed in various workstations, it is easier to model the scenario by instructing software agents to visit

and interact with each of these workstations according to the itinerary defined by the workflow. The characteristic of software agents brings a more intuitive and powerful mechanism by changing from RPC (remote procedure call) model to RP (remote programming) model when the workflow engine has to perform a sequence of remote activities that are not performed by human. The mobility characteristic also helps the coordination of workflow participants. In addition, the intelligence and pro-activeness of software agents can provide timely help for workflow participants in the workspace as shown in Figure 29.

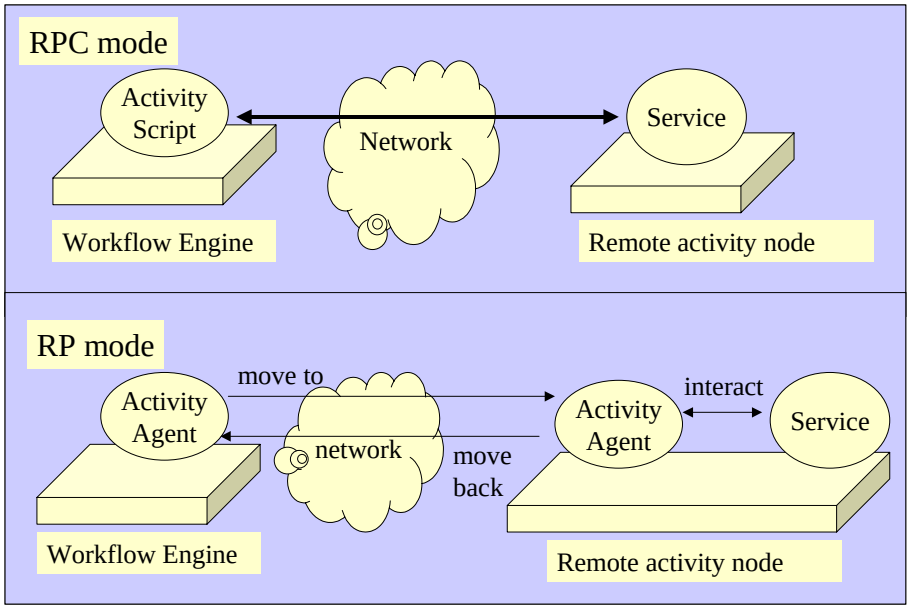


Figure 29. Modeling remote workflow activities via RPC and RP

Although domain-specific workflow applications can be empowered by deploying special-purpose software agents such as information-gathering agents or price-negotiating agents, it is not the main topic here. Our presentation is focused only on the enrichment of software agents that are directly related to the workflow platform and workspace. Thus, the software agents in the workflow system make effort on monitoring the operation of workflow engine and the interactions of workflow participants. Software agents employed in here need more intelligence than mobility, so that most of the software agents mentioned in the chapter are stationary agents (i.e. agents that never migrate).

A software-agent used for workflow systems consists of traditional software modules as described in the WfMC workflow reference model [2], and a group of software agents watch the activities of workflow engine and participants. At the server sides, agents proactively analyze the workflow run-time repository with their knowledge to instruct the workflow engine to make smarter routing decision. For the human participants, agents can provide more process predictions and decision advices for the current workflow activities according to the personalized business intelligence rules.

The execution places of agents that directly affect the human workflow participant are decided by the following criteria: (a) Agents are placed at the server side if they need to access a large run-time repository. The suggestions for the agents that are relevant to the end users are displayed through the Web pages of the enterprise process portal. (b) Agents are placed at the end user's computer if they have to remind the user with local multimedia resources or need to integrate the local access-control token with the enterprise process portal.

The portal technology is widely adopted by the enterprise for several years. Most enterprise portals are presented to the end users through Web. An enterprise information portal (EIP) usually provides end users with organized hyper-links to enterprise resources, documents browsing and search functions, and office automation applications such as mail services, address books, calendar, to-do lists, discussion groups, etc.

The enterprise process portal (EPP) is a portal that provides user interfaces for general workflow operation, and augments traditional portal applications with relevant process information. Contemporary portal products usually equipped with *portlet* [29] or equivalent technology, such as *sitemesh* [30], to help modular Web page composition and personalize Web page layout. With the ability of modular and personalized page composition for end-user screen, workflow system can provide rich process intelligence based on the end-user's preference.

A sample software agent is the *decision analysis agent* that analyzes the user's decision history of the same type, and takes current application data as input parameters (for example, the payment amount and the position of applicants in the payment approval process) to suggest the decision on the application screen, and/or make the decision as the default action when the end-user intends to submit the activity and continues next step without any comments.

Another sample is *batch approval agent* that performs activities for the user by pre-defined rules. When the user enables his batch approval agent, the agent automatically searches the activities matched by the rules, summarizes and aggregates the matched activities into single activity, and prepare default decisions defined by the rules. The end user can then make decision for routine works by completing the single activity without separate operations on each non-aggregated activity.

For the user with a long list of task activities in the workflow system, the *activity arrangement agent* provides smart suggestions to the user in addition to the plain display options such as list activities by the order of priority, due days, or other static attributes. The agent has the knowledge of average process time for several types of activities, monitors on-line users in the process portal. Therefore, it can provide a smart order for an activity list based on the criteria requiring dynamic process knowledge. For example, the user can make a choice between '*simple activity first*' or '*hard activity first*' (e.g., simpler activity means less process time on the activity, and satisfies the short-job-first schedule heuristic).

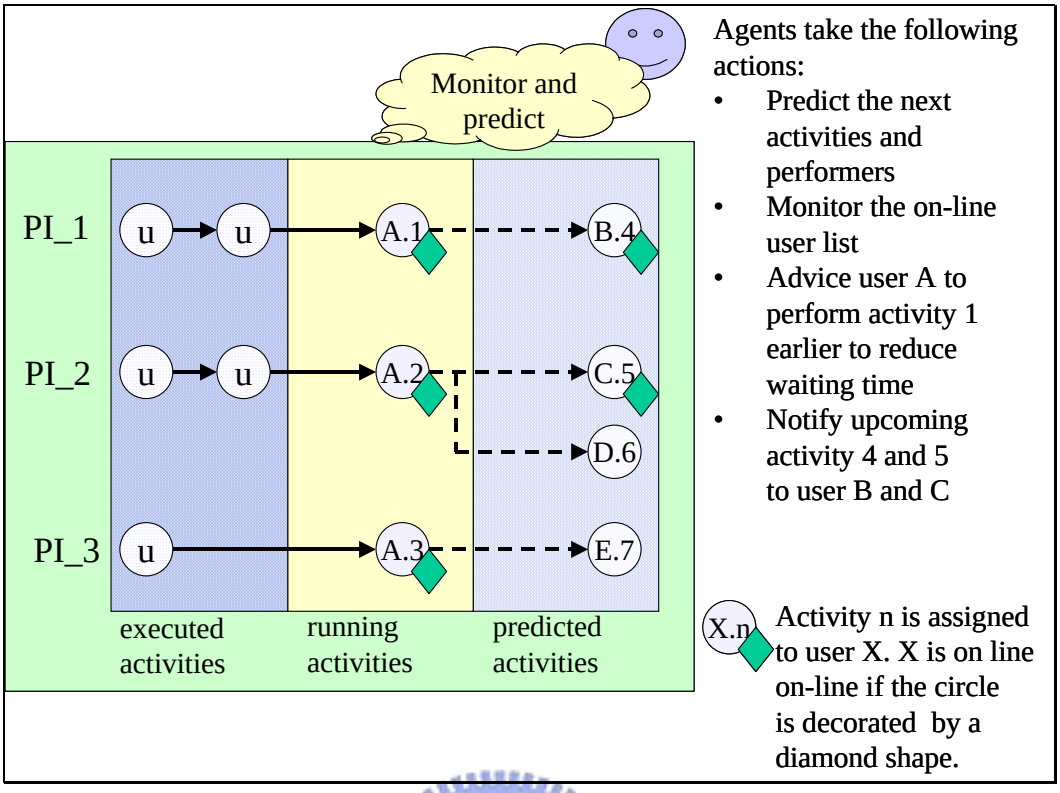


Figure 30. Agents monitor workflow run-time and predict activities to help reduce waiting time.

Based on the on-line information of the portal, an agent can compute the activities, whose next performers (if predictable) are on-line, then label and move them to the beginning of the activity list. The agent for the next performer of the activities can also keep the process knowledge to answer the question about possible upcoming activities. The scenarios demonstrate the intelligent software agents that provide useful activity information to predict and link the performers between consecutive activities. With the collaboration of activity performers and agents, the turn-around time of the whole process can be reduced because the next performer is on-line to perform the next activity with minimal waiting time.

The agents of these classes can be put into the enterprise process portal incrementally by increasing the configuration of the portal and workflow system, because the independent suggestions for software agents of different classes can be displayed in separate *portlets* of the Web page. The arrangement reduces the efforts required for the business process consultants

to inject their ideas into the software agents. Thus, the workspace containing injected business process intelligence can help users to improve their efficiency.

6.2. The Features of Agent-Empowered Process Portal

The general features of the Web-enabled workflow run-time environment are provided by three building blocks (a) workflow engine, (b) web servers, and (c) supporting utilities at user's computer. In the session we discuss the agent-empowered features for the three categories of software agents:

1. *Personal agents* work in end user's computer.
2. *Portal agents* work on the Web and portal servers as the information bridge between workflow platform and traditional enterprise portal applications.
3. *Workflow agent*s enhance and support the decisions of the workflow engine.

The mission of these software agents is to contribute the business intelligent they carry. Therefore, these agents must know well about the definition and operation status of business processes. The intelligence and capability of the software agents discussed in the following sub-sessions come from the characteristics they possess:

1. They know the process definition, including (a) the whole picture of the process definition, (b) the operational artifacts and the performer requirements for each activity.
2. They can access the run-time repository of processes and application data.
3. By the nature of software agents, they are active objects in the system. When the users are not on-line, agents act for the users; when the users are on-line, agents proactively provide timely advice or hints during operation.
4. The agents perform tasks based on the global rules and policies as enforced by the BPR experts. Meanwhile, users can define their personal preference on their agents still as long as the rules and policies are not violated.

5. Software agents play different roles according to the knowledge they carry. They can be tutors for the beginners on business operation or secretaries who filter manager's activities and inform critical activities in advance. They can also act as process advisors who provide decision support information by analyzing the process run-time repository.
6. Agents can notify workflow events or share process-related information with other agents.

6.2.1. Personal Agents

The personal agents work for workflow participants and support the daily operations of the workflow. They may work in the background of user's desktops, or work as the representative of the users on other servers. The following are some typical personal agents:

1. *Notification agent* starts automatically to stay in the Microsoft *Windows System Tray* or the equivalent places in other Unix Window Desktops. It is similar to other popular System Tray applications such as Microsoft MSN or ICQ. The agent responses for user notification and message services. They can also easily display working list status or notify the arrival of new activities automatically. Therefore, user does not need to check the activity arrival periodically in person.
2. *Sign-on agent* can simplify the login procedure by assisting the users to sign-on the system via several kinds of security check mechanisms. For example, they handle a single sign-on activity in Microsoft Windows environment. They may also handle the information stored in the security tokens such as *iKey* (http://www.rainbow.com) or other smart card products.
3. *Process instantiation agent* can prompt the users to perform scheduled activities such as to initiate a project planning process at the beginning of each season, or to initiate a working summary process bi-weekly. It can also be smart enough to

initiate a business trip report process for the user after the end of the trip by monitoring the user's business trip calendar.

4. *Process tracking agent* can monitor and notify the progress of user-assigned process instances by actively monitoring the processes in question. When the monitored processes arrive at the milestones defined, the user is notified through the *notification agent*.

Figure 31. A typical notification agent that works on the end user's PC

6.2.2. Portal Agents and Personalization

The portal agents work on the portal and web servers to gather the run-time information of the user community, and provide workflow information to other portal applications. The followings are some typical portal agents:

1. *User availability agent* monitors the sign-on and sign-off of users and maintains the history. The agent can help other agents to make their decision by providing the name list of the workflow participants.

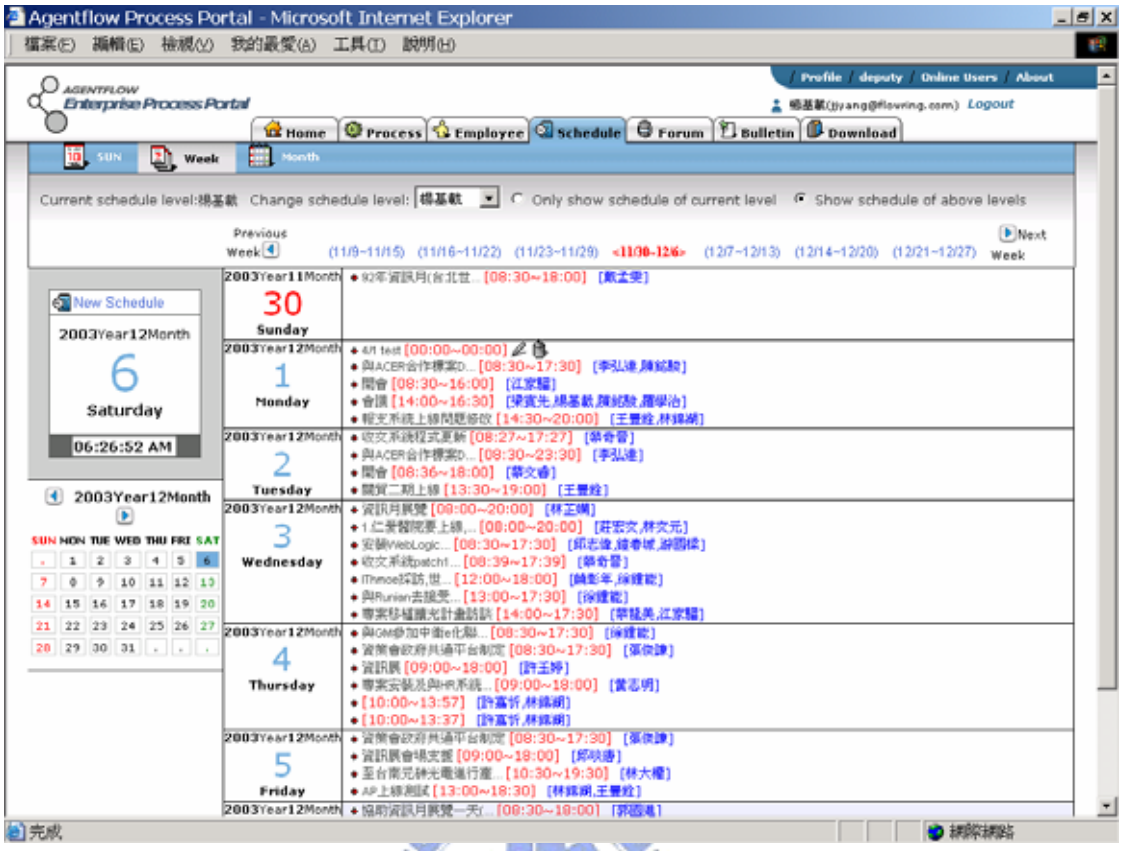


Figure 32. A calendar portlet displays the daily activities by integrating process-related information.

2. *Workflow integration agent* interacts with other portal applications on the portal servers. It provides an easier and more simple and efficient integration mechanism. For example, the *to-do list portlet* can ask the agent for work items as part of output for to-do list. The *calendar portlet* can ask for them same information, but to display the work items in terms of calendar view. The agent also performs pre-fetch and cache of workflow information to reduce both the load of workflow server and the response time of end-user's request.

3. *Work list aggregation agent* aggregates work items from several workflow engines to compose the final work list for the users. Workflow applications are sometimes divided into several servers for the reason of management or constraints of the network architecture.

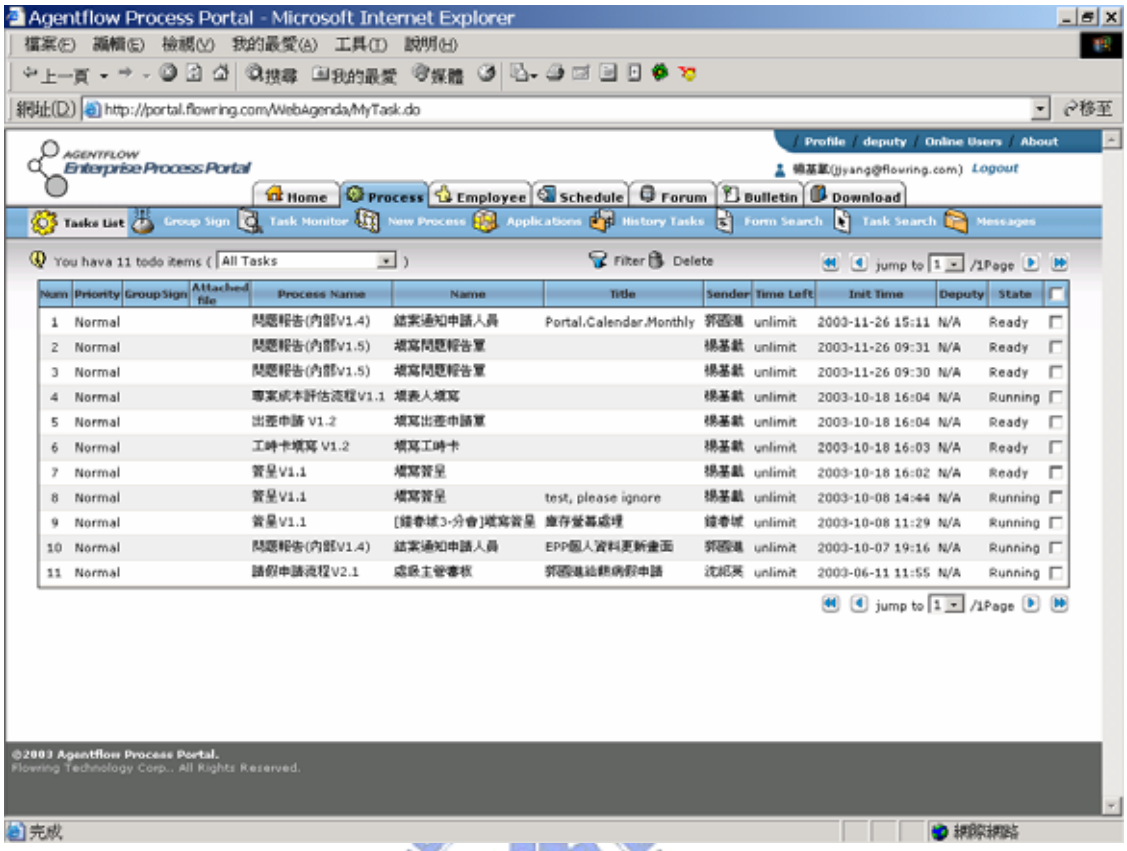


Figure 33. Work list view, which can be enhanced by agents

In a portal system the composition and style of a Web page can be personalized. The feature enables the software agents to advice the end-users through portlets. The end-users can compose the final web page by choosing desired portlets, display options, screen locations, and styles. The enterprise process portal system can provide lots of portlets (a few of them handle the advices from the agents) for users to choose.

For example, the administrative staffs can choose portlets that dynamically display summary of business performance. The team leader may want to configure his workspace to oversee his team by placing a portlet that aggregates all work items of his team.

The portal and the backend workflow platform are used together to bring new possibility for the enterprise's workspace. Users can enjoy the unified workspace for traditional Web applications and workflow applications. Users with various business roles can configure their workspaces to meet the respective duty requirements. They can also get benefits from the advices, based on richer information workflow in their working context.

Figure 34. A sample screen for the enterprise process portal.

6.2.3. Workflow Agents

The *workflow agents* can work in parallel with the workflow engine to enhance the operation capability of workflow engine. There are several application scenarios that software agents can work more powerful and transparent than traditional workflow products do. For example, they can handle advanced workflow routing issues that can be best resolved by external knowledge. The following are some sample workflow agents:

1. *Smart dispatch agent* that dynamically finds the proper candidate performers of an activity according to the agent's knowledge about the skill set of each performer and the required skills for activities.
2. *Activity queue coordination agent* monitors a set of shareable activities and their potential performers to decide better binding on the performer of each activity. Traditional approach usually binds the performer to an activity very early under the random, round robin, or weighted-dispatch policy. Sometimes, it also uses the

amount of activities for each candidate as the dispatch parameter. A traditional static dispatch mechanism might not be satisfied with time-critical activities, because it often comes out with longer process turn-around time due to improper performer assignments. The agent can improve its mechanism by collecting more dispatching parameters. For example, the agent can communicate with other agents for on-line candidates dynamically. The agent can also collect answers from the on-line candidates through instant and short communication with the *notification agent*.

3. *Substitute or delegation agent* temporarily transfers the user's active activities based on pre-defined rules to his job substitutes. It can handle recursive substitutes and conflict cases well because it has more knowledge about the availability of workflow users than a traditional workflow manager has. In addition, there exist multiple substitute rules to be use to delegate the activities to different people based on the attributes of activities.
4. *Ad-hoc routing agent* manages dispatched activities that allow ad-hoc routing. It helps collect the opinions or confirmation of dynamically assigned group members. The collecting sequence can be either parallel or sequential. The agent can resort to traditional workflow user interface (for full-control), notification agent's instant message (for quicker response), or email (for members not in the workflow organization), for routing information through group members. The agent finishes the ad-hoc routing by summarizing the final result then continues prior activities.
5. *Exception handling agent* handles system exceptions based on predefined exception-handling procedures. The exceptions in the workflow system are usually cased by (i) system failures that can be recovered by system administrators, and (ii) illegal operations due to unreasonable process definitions or inconsistent changes to

business rules. The agent can react to the exception by notifying administrators, performing rollback, or restarting from the last milestone.

6.3. The Supporting Architecture for the Environment

The features and application scenarios of the software agents in the previous session can be supported by a workflow platform, where the software agents work and which facilitates the portal technology. Software modules shown in Figure 35 work jointly to demonstrate the characteristics that an '*Intelligent Process Portal*' brings (1) intelligence, provided by the software agents, (2) process power, provided by the workflow server, and (3) personalized application presentation, provided by the portal server.

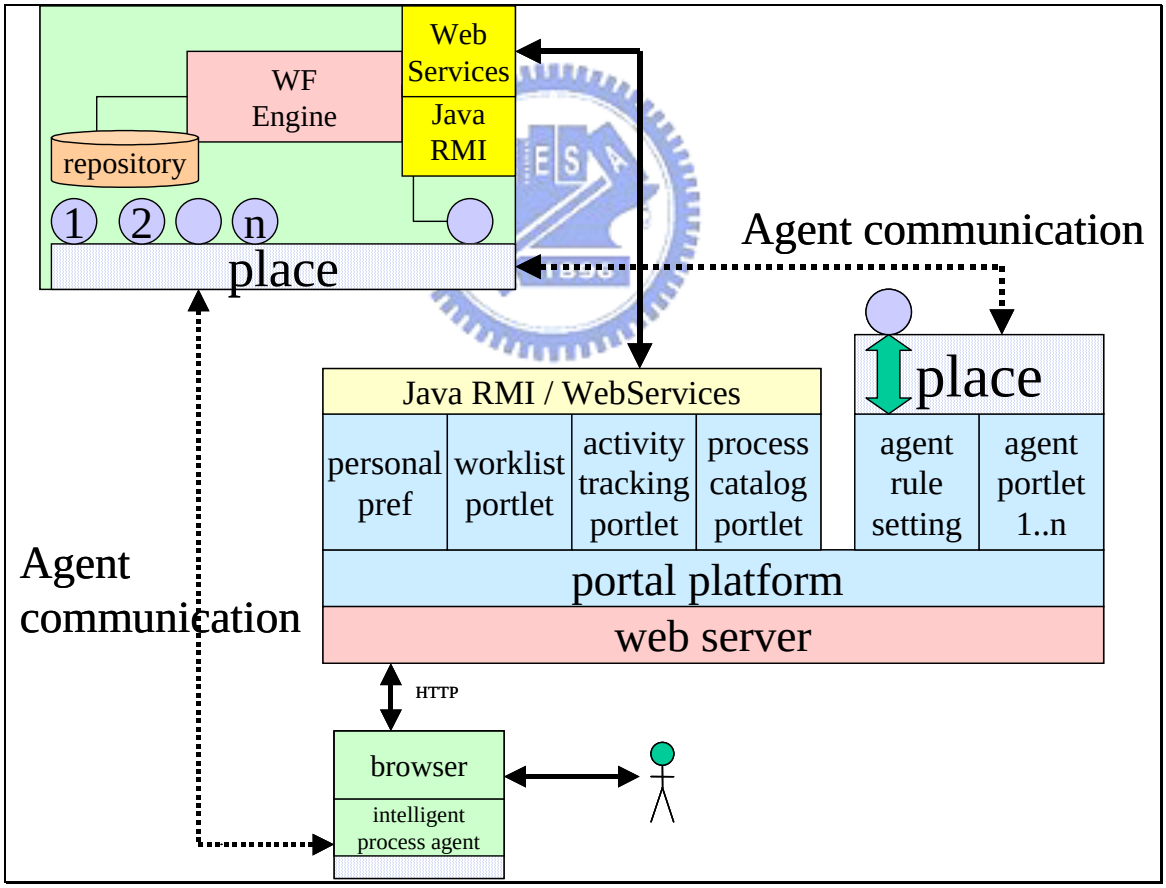


Figure 35. Architecture of the software agent enabled process portal.

The architecture is derived from traditional workflow architecture by adding the following characters:

1. Software agents are engaged at the workflow server, portal server, and the end-user's computer. Compared with a rule-based approach of the same purpose, using the software agents can lead to clearer and extendable architecture at the abstraction level. Because software agents are active objects that perform more object functions than passive objects do.
2. Agents communicate and interact with each other through their communication mechanism that is independent with the workflow system. Agents can usually work together for more complicated tasks through their communication. Agents can access workflow related information through APIs exported on the workflow servers. Workflow engine provides APIs in both Java RMI and Web Services for native language integration and cross platform integration respectively. For third-party portlets that need to access workflow information and control the workflow engine, the Web Services can be used as a more general model for cross-platform integration.
3. Traditional Web server is augmented with a portal framework that works well for aggregating several Web application output and personalized workspace.
4. Essential features on workflow user interfaces such as activity list, activity tracking, and process catalog are also provided in form of portlets to integrate seamlessly with the portal.
5. *Workflow agents* and *other agents* interact with end users through *agent portlets*. As a portlet, an agent portlet can be configured through standard portal features for layout and configuration parameters/rules of the agents.

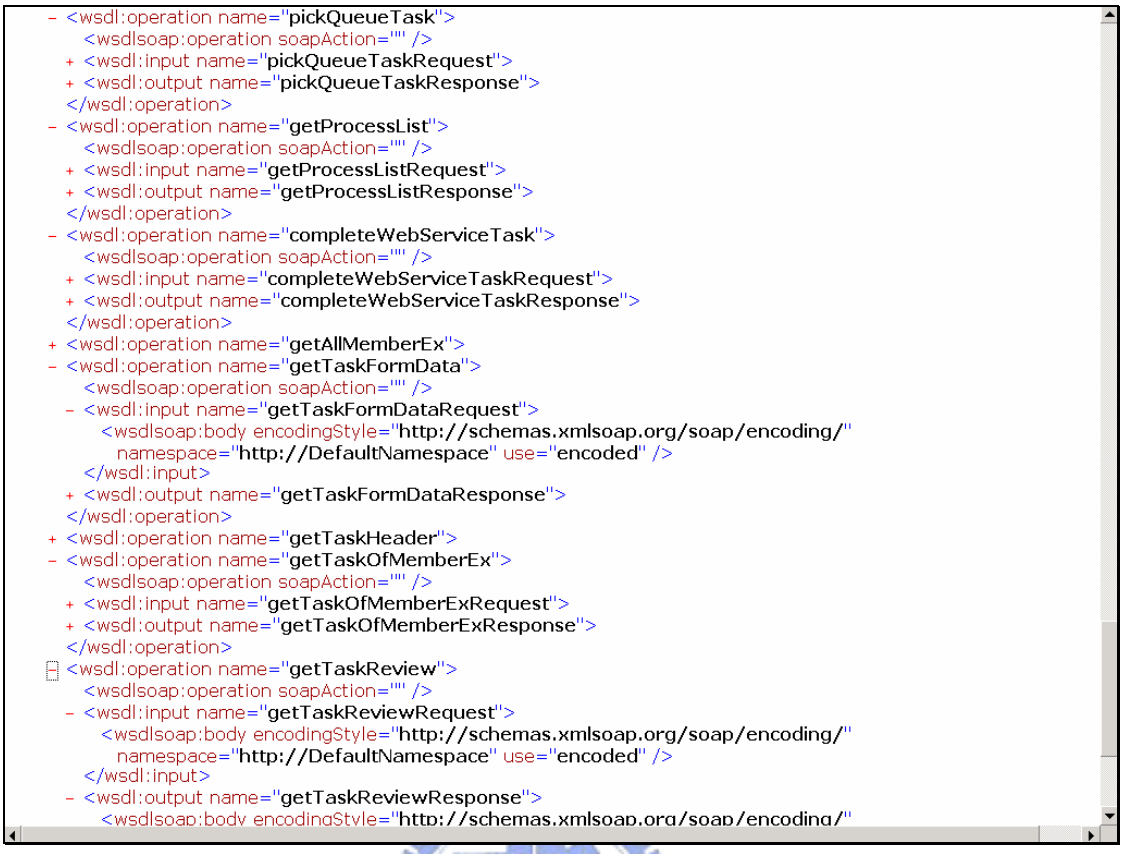


Figure 36. Third-party portlets can integrate with workflow engine through Web services.

6. At user's computer, the *intelligent process agent* is introduced to provide better workflow interaction to the local user. The agent also supports on-line chat, and instant message features. It helps the user to join proper chat channels based on his activities at hand. The agents act for users to establish a common chat channel for those who are in the same process instance. Agents also automatically enroll the users to the virtual workspace represented by the channel. In the virtual workspace, the users can chat with their working peers; attach files to the workspace as they usually do in P2P applications. The agent can record the chat history and important operations as the attachment in the process instance for auditing and tracking purposes. With the support of the *audited instant discussion*, unnecessary repetitive or compensation procedures on past activities can be significantly eliminated while actions on the process instance can be audited and

tracked. Consequently, the turn-around time of process instances can be largely reduced.

Chapter 7 Conclusion

In this dissertation, we present a cooperative workflow model CA-PLAN and an Agentflow system. The CA-PLAN is a service-oriented workflow model, where an inter-organizational process can be partitioned into several parts (co-processes) with boundary functionalities in different organizations. A co- process is defined as a process which contains a pair of process service and remote process supported by IWC connection between their process services and remote processes. An inter-organizational process is defined as a routing path and it can be assembled by IWCs through connection between their process services and remote processes. This mechanism increases modularity, flexibility and reusability in designing inter-organizational process.

The RCP provides a flexible cooperative mechanism by which the process service and the remote process communicate and pass information back and forth. Through RCP, the cooperative process across organizations becomes simple, faster, and flexible. The newly distributed Agentflow system presents the construction of an internet/inter-organization workflow so that workflow designers, working on different portions of an inter-organizational workflow systems, may work together to accomplish a common mission. It also provides a friendly tool that can be used to model inter-organizational workflow for cooperation. Also, Agentflow system provides a convenient inter-organizational process monitor to track and monitor an inter-organizational process instance.

However, the security discussed in the dissertation is a straight approach. It is necessary to be studied further. The exception handling and distributed transaction mechanism of cooperative processes among multiple WfMSs can support interoperability more transparent, sturdy, and consistent. Hence, applying excepting handling and transaction mechanisms of processes into inter-organizational workflow system are our future topics.

Besides, we present an intelligent and personalized process portal which requires an integration of workflow, portal and software agents. The ability of a workflow system to make external software modules such as software agents and portal portlets to access its process related information through standard APIs and Web services is one of the important success factors behind the integration. The process portal can enhance the intelligence by continuously updating software agent's business process knowledge. The richness of the process portal can be extended by more portlets. The overall process portal environment can also grow with its underlying information technologies from several open standard organization.

Reference

[1] D. Georgakopoulos, M. Hornick and A. Sheth, “An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure,” Journal on Distributed on Parallel Database, 3(2), 1995 pp. 119–153.

[2] WfMC Workflow Management Coalition. http://www.wfmc.org, 1996.

[3] Jie Meng, Stanley Y. W. Su, Herman Lam and Abdelsalam Helal, “Achieving Dynamic Inter-Organizational Workflow Management by Integrating Business Processes, Events and Rules,” Proceedings of the 35th Hawaii International Conference on System Sciences, 2002.

[4] M. F. Chen, B. S. Liang and F. J. Wang, “A Process-Centered Software Engineering Environment with Network Centric Computing,” Proceedings of the 6th IEEE Workshop on Future Trends of Distributed Computing Systems, 1997.

[5] W. R. Cockayne and M. Zyda. “*Mobile Agents*.” Manning Publications Co., 1998.

[6] D. B. Lange and M. Oshima. “Seven good reasons for mobile agents”, *Communications of the ACM*, 42(3):88-89, March 1999.

[7] Workflow Management Coalition, “*Workflow Management Coalition Terminology & Glossary*”, the Workflow Management Coalition Specification, Feb 1999.

[8] David Hollingsworth, WfMC, “*The Workflow Reference Model*,” Technical Report TC-1003, Jan. 1995.

[9] A. Sheth., W. Aalst, and I. Arpinar, “Process Driving the Networked Economy,” IEEE Concurrency, Vol. 7, No. 3; July-September 1999, pp. 18-31.

[10] Dickson K. W. Chiu, S.-C. C., Kamalakar Karlapalem, Qing Li, Sven Till (2002). Workflow View Driven Cross-Organizational Interoperability in a Web-Service Environment. Web Services, E-Business, and the Semantic Web, CAiSE 2002 International Workshop, WES 2002, Springer.

[11] Fabio Casati, A. D. (2001). "Modeling and Managing Interactions among Business Processes." Journal of System Integration 10(2): 145-168.

[12] Fabio Casati, Angela Discenza, "Supporting workflow cooperation within and across organizations," Proceedings of the ACM symposium on Applied computing, Como, Italy, 2000.

[13] Stanley YW Su, J. M., Raja Krithivasan, Seema Degwekar, Sumi Helal (2003). "Dynamic Inter-Enterprise Workflow Management in a Constraint-Based E-service Infrastructure." Electronic Commerce Research 3(1): 9-24.

[14] Myungjae Kwak, Dongsoo Han, Jaeyong Shim, "A Framework Supporting Dynamic Workflow Interoperability and Enterprise Application Integration," Proceedings of the 35th IEEE Annual Hawaii International Conference on System Science, 2002.

[15] Mehmet Sayal, Fabio Casati, Umesh Fayal, Ming-Chien Shan, "Integrating Workflow Management Systems with Business-to-Business Interaction Standards," Proceedings of the 18th IEEE International Conference on Data Engineering, 2002.

[16] G. Alonso et al., “WISE: Business to Business E-Commerce,” IEEE 9th International Workshop on Research Issues on Data Engineering, Information Technology For Virtual Enterprises, Sydeny, Australia, March 23-24 1999.

[17] [URL]http://lsdis.cs.uga.edu/proj/meteor/meteor.html

[18] Object Management Group, “CORBA/IIOP 2.2 Specification,” 10 Feb. 1998.

[19] H. Davulcu, S. Dawson, M. Kifer, L. R. Pokorny, C.R. Ramakrishnan, I.V. Ramakrishnan, "Modeling and Analysis of Interactions in Virtual Enterprises", 9th International Workshop on Research Issues on Data Engineering: Information Technology for Virtual Enterprises, RIDE-VE,Sydney, Australia, 1999.

[20] H. Davulcu, M. Kifer, C.R.Ramakrishnan, and I.V.Ramakrishnan. "Logic based modeling and analysis of workflows", ACM Symposium on Principles of Database Systems (PODS), Seattle, WA, June 1998.

[21] Dickson K. W. Chiu, S. C. Cheung, Sven Till, Kamalakar Karlapalem, Qing Li, Eleanna Kafeza. “Workflow View Driven Cross-Organizational Interoperability in a Web-Service Environment,” Information Technology and Management Volume 5, Issue 3-4 July-October 2004.

[22] WfMC, “Interoperability Wf-XML Binding,” Technical Report TC-1012, May 2000.

[23] Standard ECMA-262 ECMAScript Language Specification, http://www.ecma-international.org/publications/standards/ECMA-262.HTM.

[24] Flowring Technology Corp, Agentflow system, http://www.flowring.com

[25] Yin-Shinn Chen, Feng-Jian Wang: An Editing System for Working Processes. COMPSAC 2001.

[26] Sun Microsystems,” Java(tm) Remote Method Invocation (RMI)”

[27] Sun Microsystems,” Java Naming and Directory Interface(JNDI)”

[28] “Lightweight Directory Access Protocol(LDAP)”, RFC2551, 1997

[29] Sun Microsystems, “Introduction to JSR 168 - The Portlet Specification”. Via URL http:// developers.sun.com/prodtech /portalserver/reference/techart/jsr168/

[30] Open Symphony, “SiteMesh Overview,” via URL http://www.opensymphony.com/sitemesh/

Vita

Shung-Bin Yan was born on October 2.1970 in Taipei, Taiwan, Republic of China. He received the BS degree in Computer Science and Information Engineering form National Chiao Tung University, Taiwan in 1994. He works for Flowring Technology as the RD director. He is also a Ph.D candidate in computer science and information engineering at National Chiao Tung University. His research interests include software engineering, workflow system, object-oriented analysis and design and distributed software system.