AN INTER-ORGANIZATIONAL WORKFLOW MANAGEMENT SYSTEM FOR VIRTUAL ENTERPRISES

TAUQIR AMIN

NATIONAL UNIVERSITY OF SINGAPORE
2002

AN INTER-ORGANIZATIONAL WORKFLOW MANAGEMENT SYSTEM FOR VIRTUAL ENTERPRISES

TAUQIR AMIN

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

2002

ACKNOWLEDGEMENTS

I want to thank my advisor, Associate Professor Hung Keng Pung, for inviting me to this study. Thank you for your encouragement, thank you for your willingness to allocate your time and analytical skills to our discussions over the past 2 years, and thank you for your continued guidance and constructive criticism.

I would also like to appreciate other members of HISFlow group: Ms. Zhang for his HISFlow visual builder and Mr. Tariq Riaz for his help in reviewing the thesis.

Finally, the opportunity to do M.Sc. offered by the National University of Singapore is gratefully acknowledged.

CONTENTS

LIST OF FIGURES

SUMMARY

Inter-organizational Workflow Management System for Virtual Enterprises

Abstract

There is a need for organizations to have synergy by joining their local workflow processes. These processes may have not developed with the consideration of any virtual enterprise. We propose a Process Meta model to integrate the processes by introducing a new concept of Partial View of Virtual Process. Using this approach, local processes of participating organizations could be integrated with other external processes quickly. The procedure of joining and disjoining virtual enterprise is transparent. To be able to adopt this approach, architecture and implementation of a virtual process enactment service is presented. This framework is a set of collaborating intra-organizational workflow management systems nodes. We also present internal architecture and implementation of such intra-organizational workflow management system – called HISFlow. A theoretical foundation is also laid down to exploit workflow history information based on our proposed meta model for a virtual process.

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Although much of today's office work is cooperative, the automation of workflows governing the handoff and routing of business documents and data are often hard-scripted into supporting software. This tight coupling of processes and supporting applications limits the modeling, deployment, and execution of work. These shortcomings are compounded by the changes in organizational structures to accommodate dynamic and geographically dispersed workgroups. Efficient and successful business process management is one of the major challenges organizations are facing with all over the world today. Business processes are the basis upon which organizations re-organize their structures, production processes, adaptation to changing demands of customers and their relationships to other players on worldwide markets. Coordination and workflow management technology are crucial technological enablers for achieving and keeping competitive organizational structures.

The penetration of Internet and web, in accordance with new technological advances, urged organizations to seize the opportunities offered by electronic commerce and to establish a strategic position in the new global networked world. In order to achieve that, organizations should co-operate in different product development phases and share critical business processes, resources, core competencies, skills and know-how with each other [Christofer 93, Applegate96, Ouzounis 99a, Camarinha-Matos 99]. This new business model lead to the concept of virtual enterprises that is the foundation of the networked economy [Ouzounis 99a,b, Camarinha-Matos 99, Fielding98].

The original goals of the virtual enterprises are to enable deployment of distributed business processes among different partners, to increase the efficiency of existing services, to decrease the cost for these services, and to adapt to new market changes [Banahan 99, Stricker 00]. As organizations introduced electronic business systems, they started to see new possibilities enabled by them. By more closely coordinating the work of business partners, businesses see dramatic productivity and efficiency increases in their processes. As communication barriers and costs drop, businesses are able to engage themselves in many kinds of relationships [Doz 98, Adams 97].

Workflow is a computerized facilitation or automation of a business process, in whole or part. A workflow management System completely defines manages and executes "workflows" through the execution of software whose order of execution is driven by a computer representation of the workflow logic [WFMC]. Workflow management systems (WFMS) feature a set of good attributes for deployment within the context of a virtual enterprise. The shared business processes among the participating organizations of a virtual enterprise can be described by deploying a business process specification language. In that way, the shared business processes can be easily developed. For example, a shared process can start in one organization and then, can be continued in another organization, by utilizing remotely a sub-process. The WFMS will undertake the responsibility to control and manage the execution of the shared business process in a distributed and systematic way [Hoffner 98].

The workflow management systems based on classic middleware systems typically involve tight binding between the systems and processes at the various organizations [Ouzounis 98e, Thompson 99]. By closely coupling the organizations, classic middleware systems are able to provide rich functionality. However they require expensive development and deployments, and pre-agreement in the interfaces used. Therefore, these solutions are better suited for use in intra-organization or long-term business partnerships, i.e. static virtual enterprises [Ouzounis 99a, Spinosa 98].

Currently proposed standards of WFMS technology are not directly dealing with inter-organizational business process execution and management. The critical problems include inter-organizational workflow execution and management, business process specification languages for inter-organizational business processes, and dynamic selection of potential organizations. These problems are rarely discussed in the literature [Bolcer 99]. Additionally, the deployment of tightly coupled communication mechanisms, like CORBA, is rather problematic and in general, inflexible. Therefore, it is anticipated that a message-based approach with corresponding XML message requests and responses would have been better since the degree of autonomy and flexibility is increased [Georgakopoulos 98].

In order to be competitive in the dynamic marketplace, healthcare enterprises need to deploy information technology solutions that internally automate the paper-based medical record systems and externally create smart connections between the major participants in the healthcare. With the emergence of near ubiquitous Internet connectivity, a virtually integrated healthcare enterprise built upon connections, partnerships, alliances, and relationships with physicians, polyclinics, laboratories, pharmacies, hospitals and payers is emerging as the operating model for healthcare organizations [Horsch 99]. The potential participating organizations of a virtual healthcare enterprise are sufficiently rich in their infrastructure to handle the internal administrative and clinical processes [Deloitte 00]. The need to integrate the processes of such geographically distributed organizations is evident.

The most relevant activities within a healthcare organization can be described in the form of business processes as mentioned in [Maij 00]. Following this idea, we define a virtual process as a business process whose definitions and enactment are distributed among more than one organization. From here, we define a virtual enterprise as the one whose business processes are virtual processes. The potential participating organizations do have their own internal processes which are developed previously without considering that they could be part of any virtual process. Therefore building and enacting a virtual process from such local processes is a big problem. The problem is intensified when dynamics of relationships, autonomous behavior of local processes and heterogeneity of WFMSs are considered. To solve this problem, changes in process meta model and process enactment framework are required. The changes both in process meta model and process enactment framework should be such that they could be easily adopted by the WFMSs that are following standards.

Quite a few attempts have been made to develop a WFMS for healthcare sector. Most of the currently available WFMS solutions are too complex [AdDadam 00] and hence too costly. In the last few years, pervasive network connectivity, explosive growth of Internet and web technologies have changed our computational landscape from centralized, desktop oriented and homogenous computing to distributed, heterogeneous and network centric computing model. These emerging technologies have raised new opportunities to simplify the core of workflow management system. HISFlow - a Workflow management system for healthcare is an attempt to exploit these new technologies, mainly Enterprise Java Beans (EJB), Java Server Pages (JSP), XML and Service-oriented Programming Model.

1.2 TARGETED PROBLEMS

In this dissertation, we discuss a bundled solution to address the problems related to inter-organizational workflow management in the context of virtual enterprises. The different problems we tackle are

1. To enable the organizations to design their intra-organizational workflows quickly and graphically.
2. To enable the organizations to enact and monitor their intra-organizational workflows.

3. To enable the potential organizations of a virtual enterprise to build the virtual processes by joining their already developed intra-organizational workflows.
4. To enable the participating organizations of a virtual enterprise to enact the virtual processes.
5. To keep the dynamics of relationships, the autonomous behavior of intra-organizational workflows, and the heterogeneity of WFMSs of participating organizations of a virtual enterprise intact while designing or enacting the virtual processes.
6. To let the organizations have a design approach to transform their intra-organizational workflow management system to inter-organizational workflow management system with minimal changes required.
7. To enable the organizations to exploit the workflow history information for various purposes in the context of a virtual enterprise.

1.3 CONTRIBUTIONS

1. We exploit the latest advancements in the middleware and Internet technologies to keep the design and implementation of an intra-organizational workflow management simple and cost effective unlike most of the currently available WFMS solutions that are too complex and hence too costly for Small and Medium Enterprises (SMEs).
2. We propose a new approach of partial views to model a virtual process that lets those intra-organizational workflows join together, which were developed without consideration of a virtual enterprise.

3. We demonstrate how an already developed intra-organizational workflow management system can be transformed to the one which can take part in a virtual process enactment framework with minimum changes made.
4. We lay down the theoretical foundation by introducing a scheme to identify workflow history information so that it can be exploited in the context of virtual enterprises.

1.4 OUTLINES OF THESIS

We divide the thesis into eight chapters that present the state of the art in the area of workflow technology as a solution for the virtual enterprises. It includes the specification of process meta models for both intra-organizational workflows and inter-organizational workflows, the architecture of intra-organizational WFMS, the meta model for the virtual processes, and the architecture of virtual process enactment.

More specifically, the chapter 2 presents the literature review that includes classification of workflow and workflow systems, the concept of a virtual enterprise, the related work done in inter-organizational workflow models and, different categorization and approaches of architecture of workflow management system.

The chapter 3 gives the overview of HISFlow, an intra-organizational workflow management system. It covers the key definitions and concepts related to process meta model, role model, and application model. An illustrative example is also given to explain the application of the models.

The chapter 4 presents the virtual process meta model for inter-organizational workflows based on the proposed concept of partial view of a virtual process. A real world scenario of four organizations is used to further elaborate different aspects of the model.

After having presented the process meta models for both intra-organizational workflows and inter-organizational workflows in the previous chapters, the architecture and implementation of the intra-organizational workflow management system is discussed in the chapter 5. This includes user view of the system, system architecture, and the subsystems: build-time subsystem and run-time subsystem.

Based on the intra-organizational workflow management system, the architecture and implementation of virtual process enactment framework (inter-organizational workflow management system) is proposed in the chapter 6. The chapter also includes the design of extended workflow engine, meta model of virtual process enactment, and build-time and run-time interactions of coordinating workflow management systems.

The chapter 7 discusses another capitalization of the workflow technology in the context of a virtual enterprise which is workflow history information. After summarizing the different applications of workflow history information, this chapter only proposes the solution of one of the problems of workflow history management i.e. identification of workflow history information.

Finally, the chapter 8 presents the conclusion drawn from the work done in the thesis. More specifically, it discusses advantages and disadvantages of the solutions of targeted problems. It also describes the open issues for future R&D work. This description can function as a motivation for future and more extensive research in the area of inter-organizational workflow management system for the virtual enterprises.

CHAPTER 2

REVIEW OF LITERATURE

Although the term workflow management system is used first time in late 80s but quite a time ago sub-set of same kind of issues, functionalities and challenges were mentioned and researched in some form or the other but under different names. This name list includes Imaging System, Document Management, Electronic Mailing System, Process Automation, Groupware Applications and Business Process Re-engineering (PBR). The evolution of these concepts collectively can be taken as evolution of workflow management technology. In this chapter, we identify the research efforts and highlight the representative research results achieved in the following five areas: (1) Classification of Workflows, (2) Virtual Enterprise, (3) Inter-organizational Workflow Models, (4) Architectures of Workflow Management Systems, and (5) Technologies

2.1 CLASSIFICATION OF WORKFLOWS

Up till now there exists no general accepted classification framework for workflows (processes) and workflow systems. Since every classification focuses on some specific aspects, it will always be difficult and probably impossible to give a commonly accepted classification. In the literature, there are several attempts to classify workflows and workflow systems [Georgakopoulos 95, Marshak1994, Marshak 1997b, Abbot 1994, Silver 1994, Ader 1997]. We briefly review some of these classifications.

2.1.1 Georgakopoulos' Classification Scheme

In [Georgakopoulos 95] workflows are characterized along a continuum from *human-oriented* to *system-oriented workflows*. In the first case, a workflow is mainly performed by human agents. The WFMS is expected to support the coordination and collaboration of humans who are responsible for consistent workflow results. In the second case, workflows are characterized as highly automated and computation-intensive processes which involve the integration of heterogeneous, autonomous and / or distributed information systems. Since human influence is very limited, system-oriented workflows must include software for various concurrency control and recovery techniques to ensure consistency and reliability.

2.1.2 Abbot and Sarin's Classification Scheme

Abbot and Sarin's scheme [Abbot 1994] for workflow product characterization spans out three orthogonal dimensions: *mail-driven vs. database-driven*, *design vs. runtime* and *document-oriented vs. process-oriented*. The authors characterize their own product, *InConcert*, as runtime, database driven and process-oriented.

2.1.2.1 Mail-driven vs Database-driven

Mail-driven workflow technology relies on electronic mail (e-mail) systems to support the routing (transport) of work (documents). By relying on e-mail, intra-organizational and inter-organizational interoperability are achieved, but important limitations of such an approach are:

- An inflexible representation of routing logic (relying on, and represented on, each involved e-mail client application).
- Lack of support for capturing aggregate historical information about business processes.

Database-driven workflow technology is based on access to documents and routing logic through a shared database. Hence, documents will be available all the time, they are not physically "flowing", and limitations of the typically "low-end" mail-based products are relaxed. Especially, the ability to capture aggregate information based on previously run process instances is important for control as well as for business process improvement.

2.1.2.2 Design vs Run-time

Design oriented WFMS focuses just on process modeling whereas a run-time oriented WFMS also enacts the processes. In Design-oriented WFMS, the accompanying run-time tools, like workflow engines, either are not planned to be deployed in the organization or they do not exist for the particular modeling paradigm (the latter is often the case with simulation tools).

2.1.2.3 Document-oriented vs. Process-oriented

Document-oriented workflow tools typically denote the "low-end" part of the market, built on top of e-mail products. These tools typically associate routing information directly with data objects.

Process-oriented tools (which typically are database-driven) model business processes as sequences of steps (activities), i.e. atomic actions, where data-objects are associated with these steps. This is a crucial distinction, if routing is limited to documents only, the workflow system does not actually support the "flow of work", but is limited to "flow of documents" only.

2.1.3 Silver's Classification Scheme

Another classification scheme involving actual products and a view of directions in the market place is given in [Silver 1994]. Here, a division is done in two dimensions: *production vs. ad hoc* and *document-centric vs. group-centric*. Ad hoc tools are those where workflow support is integrated with other groupware applications and in a classic sense there is no explicit workflow management system. The category of group-centered tools are those deployed to support autonomous groups of people working together to achieve some goal, who are in need of continuous communication and interaction. Document-centered workflow systems just focus on the "flow of documents" and do not necessarily take a process perspective of workflow modeling.

2.1.4 Ader's Classification Scheme

- Ad-hoc workflows do not have a well-defined process model to follow. The execution path is more or less determined at run-time, and is basically controlled by humans. These are generally not mission-critical, and accomplish the flow of information among people within an organization.

- Production workflows are also predictable and repetitive. They have well defined process models. These usually involve a number of information systems that may be heterogeneous and distributed. Production workflow management systems are thus more complex and critical than ad-hoc or administrative.

- Administrative workflows are based on simple, repetitive and predictable processes. The ordering and coordination of tasks can thus be automated. However, these too, like ad-hoc workflows, do not involve complex information processing systems and are generally not mission-critical.

- Collaborative workflows are characterized by high mission criticality. They are mostly controlled by humans and lack a well-defined process model. Therefore most of the task ordering and coordination are determined at run-time by the workflow participants.

2.2 VIRTUAL ENTERPRISE

The virtual enterprise is not a new concept in management studies [Malone 91, Adams 97, Camarinha-Matos 99a, 99b, 99c]. This research area is growing and multidisciplinary one. So far, there is no unified definition for this paradigm and a number of terms are even competing in the literature while referring to different aspects and scopes of a virtual enterprise [Bolcer 99, Carr 96, Ouzounis 99a, Filos 00, Alonso 99]. For instance, Byrne says, "a virtual enterprise is a temporary network of independent organizations, customers, even rivals linked by information technology to share costs, skills, and access to one another's hierarchy" [Byrne 99]. To Walton and Whicker "a virtual enterprise consists of a series of cooperating ‘nodes’ of core competencies which form into a supply chain in order to address a specific opportunity in the market place [Walton 98].

The wide variety of different networked organizations have led to the generation of a number of related terms such as extended enterprise, virtual organization, networked organization, supply chain management, or cluster of enterprises. Some authors use some of these terms indistinctly to the virtual enterprises although there are differences between their detailed meaning [Zarli 99, Wognum 99].

Although there is no strict academic definition regarding the virtual enterprise, different virtual enterprise models feature common business and technical characteristics and attributes. The most important features of the virtual enterprises are:

- More than one independent organizations are involved in providing the service
- The service is performed by sharing business processes and resources
- The sharing of processes and resources lasts for limited period of time
- The process interfaces might be static or dynamic.
- The number of participating organizations might be either fixed or variable according to the needs and requirements of the organizations involved.
- The participating organizations are physically distributed and are connected with electronic means and systems

Based on the above common features that the virtual enterprises have, two well-defined categories can been identified [Zarli 99, Gibon 99], namely the static virtual enterprises (SVEs) and the dynamic virtual enterprises (DVEs).

In the static virtual enterprises, a set of organizations is linked together in a static and fixed way, i.e. the shared business processes are tightly integrated. The business relationships among the organizations, i.e. the process interfaces are pre-defined, tightly coupled, fixed, and well integrated. The network is fixed and pre-determined and thus, the structure of the virtual enterprise is static and pre-determined.

In dynamic virtual enterprises a set of organizations is linked dynamically, on-demand, and according to the requirements of the customers [Wognum 99, Mitrovic 99]. The participating organizations do not have fixed business relationships and thus, the virtual enterprise is not static and might change continuously. The organizations that want to form a virtual enterprise relationship can register offers on one common place in relation to the process definitions. The partner selection procedure between the organizations is usually performed through negotiation. The negotiation procedure might be either manual or automatic [Filos, Geppert 98].

2.3 INTER-ORGANIZATIONAL WORKFLOW MODELS

2.3.1 Interoperability

Interoperability among the organizations participating in the inter-organizational workflows can be achieved in various ways. Van Der Aalst presents following forms of interoperability [Van der aalst 99].

2.3.1.1 Capacity Sharing

In this type of models, there is only one central copy of process definition controlled by one centralized WFMS. The execution of tasks is distributed involving the resources of more than one organization and hence the autonomy of the organizations is not preserved.

2.3.1.2 Case Transfer

Each organization has the same copy of process definition i.e., the process definition is replicated. At any time, each process instance (case) resides at exactly one organization. The process instance transfers to the organization when the resources of that particular organization are required

2.3.1.3 Extended Case Transfer

Same as "Case Transfer" except it is not assumed that each of the organizations uses the same process definition i.e. at a specific organization the process may be extended with additional activities (tasks).

2.3.1.4 Sub-contracting

One organization subcontracts the sub-processes (sub-workflow) to other organizations. For the top-level process the subcontracted processes are atomic. The definition of each of the sub-processes is local

2.3.1.5 Loosely Coupled

A set of local process definitions residing on the different organizations interact each other to form one global inter-organizational workflow. The local process instances may be synchronized with each other while running in parallel.

2.3.2 Loosely Coupled Inter-organizational Workflow Models

We categorize approaches of loosely coupled inter-organizational workflow models into three categories: *split and deploy*, *composition,* and *black box*. In split and deploy, a whole process is designed and then it is split into several sub-processes for deployment in different organizations. This approach is typified by METEOR project [Aalst 99] and Public to Private [Aalst 01]. In the approach of composition, a whole process specification is obtained from several segments that may be contributed by different organizations. Then the process is built and deployed. WISE [Lazcano 00] and CMI [Georgakopoulos 99] take this approach. A good representative work of this approach is process fragments [Lindert 99]. The black box approach introduced by CrossFlow [Grefen] lets the participating organizations specify their collaboration in a central electronic contract. Based on this contract, services are configured and outsourced among the participating organizations of a virtual enterprise.

In the CMI [Georgakopoulos 99], a project at MCC, inter-organizational workflow was achieved by using a Collaboration Management Model (CMM) sitting on top of other workflow models to map methods and tools for defining services provided by different organizations. The basic idea of CMI is the integration of processes via inter-organizational services. CMI does not aim at designing a new workflow management system. They assume that processes are enacted by a commercial WFMS instead. In WISE [Lazcano 00], a web based e-commerce platform project, the workflow designers can post their design segments into a common World Wide Web (WWW) based catalog repository. A virtual process specification can be constructed by using the segments retrieved from the catalog. This specification can then be compiled and the resulted processes will be enacted by the WISE engine. This published work does not however present a concrete process model for the specification of cooperating processes, and does not show how the interaction is achieved.

2.4 ARCHITECTURES OF WFMS

2.4.1 Miller and Sheth's categorization

Miller and Sheth [Miller 96] present five categories of WFMS implementation architectures covering Highly Centralized, Synchronously Centralized, Asynchronously Centralized, Semi-distributed and Fully-distributed architectures. It also discusses the advantages and disadvantages of each category.

2.4.2 Kim's Taxonomy of WFMS's Architectures

Kim gives another and more general classification of architectures of WFMS in [Kim 01]. He presents a three level workflow architectural framework that includes *generic*, *conceptual* and *implementation* levels.

2.4.2.1 Generic Level

For generic level, it categories WFMS based on three dimensional criteria – *Control*, *Application Data* and *Execution Scripts* (Applications). Each of these dimensions is further divided into *Centralized*, *Decentralized* and *Dispersed*.

Centralized: control, application data and / or execution scripts (Applications) are kept at a single site. A site is a node of the workflow network.

Decentralized (replicated): full copies of control, application data, and / or execution scripts (Application) are distributed to more than one site.

Dispersed (partitioned): control, application data, and / or execution scripts are divided into multiple partitions, and the various partitions are kept in a non-replicated fashion at different sites.

Control dimension: concerned with scheduling or making of decisions i.e. which activity or activities should be enacted when, and by whom.

Data dimension: concerned with storage and manipulation of all the data pertaining to applications used in the workflows.

Execution dimension: concerned with execution of scripts – program segments that perform various activities within a workflow process.

2.4.2.2 Conceptual Level

For conceptual level, the architecture can be described based on what kind of concepts the WFMS deals with; the relationships of these concepts and how these concepts are materialized in the WFMS. The concepts include *Workcase*, *Activity*, *Role*, *Actor* and etc. These concepts can be materialized in a very passive way i.e. represented just as a set of data in a WFMS. Opposite to this, the concepts can be materialized in a very active way i.e. as software processes or agents.

2.4.2.3 Implementation Level Architecture

The implementation level architecture deals with caching, multiprocessing, threads versus process, network protocols, performance of the architecture, inter-network configuration and etc.

2.5 TECHNOLOGIES

The first attempts to realize inter-organizational business systems have been done by using Electronic Document Interchange (EDI). But certain problems regarding the standard format of EDI messages, the insecure open transport networks, and the rather restricted context of EDI made EDI not an attractive solution for the virtual enterprises [Billington 94, Christopher 93, Bolcer 99, Doz 98].

A number of other technologies approach inter-organizational relationships by using middleware [Stricker 00, Georgakopoulos 98, Fielding 98]. The middleware is a layer of integration code and functionality, which allows multiple distributed systems to be deployed as though they were a single system. Using these middleware services applications can transparently access the multiple, backend, distributed, legacy systems and applications [OMG 98, EJB 99, Orfali 96].

Classic middleware systems typically involve tight binding between the systems and processes at the various organizations [Ouzounis 98e, Thompson 99]. By closely coupling the organizations, classic middleware systems are able to provide rich functionality, but require expensive development and deployments, pre-agreement in the interfaces used, and carefully coordinated, ongoing deployment management.

In contrast to classic component based systems, inter-organizational systems can be built using exchange of documents, usually described in XML [Sheth 97 and 98]. Ideally, such an approach would combine the strengths of EDI with the rich interaction, integration, and distribution supported by classic, distributed component-based systems [Stricker 99]. The messaging approach of “fire and forget” seems to be better in the area of inter-organizational communication and co-ordination in comparison with the classical distributed object oriented concepts due to the loosely coupled approach.

Finally, another emerging technical area for the development of business processes is the concept of intelligent mobile agents [Magedanz 99, Krause 96 and 97]. The success story of agents started in the early nineties with the parallel appearance of different agent concepts and technologies. These technologies can be roughly separated into intelligent agents and mobile agents [Maes, 94]. The interest in agents was coined by the increasing notion of Multi Agent Systems [MAS] in the early nineties, driven by the Distributed Artificial Intelligence (DAI) research community [Wooldridge 95].

CHAPTER 3

OVERVIEW OF HISFlOW:

AN INTRA-ORGANIZATIONAL WFMS

Intra-organizational workflows are concerned with the automation of procedures where documents, information or tasks are passed between the participants of the organization according to a defined set of rules to achieve or contribute to an overall business goal. Whilst workflow may be manually organized, in practice workflow is organized within the context of an IT system to provide computerized support for the procedural automation.

In this chapter, we mention key concepts and definitions pertaining to the workflow technology and our implementation of intra-organizational WFMS, HISFlow, on which inter-organizational WFMS is based. This includes process meta model, role model, application model and process instance. Our methodology of presenting these concepts is through an illustrative example of a workflow for managing diabetic patients in Pediatrics Clinic of National University Hospital - Singapore, the collaborator of the research project. From time to time, as we move along through the chapter, we use this example to render above mentioned concepts.

3.1 SCENARIO OF DIABETIC PATIENT MANAGEMENT

Healthcare delivery is highly process-driven [Maij 00]. There are well-defined steps in the case of any diagnostic or therapeutic procedure. The completion of a step often leads to the beginning of another step and each step may require some authorization and invocation of legacy applications. Moreover, changes of diagnostic procedures due to the new health policies are common in healthcare domain. All these factors demand us to develop a WFMS that provides computerized support for the healthcare procedural automation. With such a WFMS, activities or processes captured in the system can be streamlined and their performance can be improved upon. The system can also automate collaborative work among the users and allocate resources and work according to the workload of users. Doctors can make faster and better decisions and time required to complete a task can also be shortened.

Diabetes is a life long condition which requires continual consultation [Diabetes]. The scenario under consideration is management of diabetic patients in Pediatrics Clinic of National University Hospital. When such a patient visits the clinic, he/she gets registered at the counter with the registration nurse. If this is his/her first visit, he/she has to proceed to the educator for getting his/her history information recorded, before going to the medical nurse. Otherwise he/she is asked to directly go to medical nurse who performs some of the general medical tests. If the results are ok, he/she can proceed to the doctor, otherwise the patient has to go through further medical tests in the special tests laboratory. This laboratory registers the patient again for its record. Here, eye examination and blood test are conducted. After completion, the patient is asked to go for consultation with the doctor. The same procedure is repeated for every visit of all diabetic patients.

3.2 KEY CONCEPTS AND DEFINITIONS

3.2.1 Process Meta Model

As we can see from the above scenario, the diabetic patient management at the clinic involves certain process or procedural steps. For automation of such a clinical or an administrative process, we have to capture and then transform this process into computer understandable representation. We need a process meta model for the representation of the process. Keeping in mind the standardization efforts for such process meta models, we surveyed different available process meta models including Workflow Management Coalition (WFMC) Interface 1 [WFMC 1], Business Process Modeling Language [BPMI], XLANG [Microsoft], EDOC [OMG 01], WSFL [IBM] and UML2.0 [UML]. Out of these meta models, we opt WFMC's Interface 1 (or Process Definition) for its maturity and being industry standard for WFMS. The process definition contains all necessary information about the process to enable it to be executed by the workflow enactment service (workflow run-time system). This includes information about its start and complete conditions, constituent activities, and rules and conditions for navigating between them. It also includes participants to perform these activities, references to applications that might be invoked, and definition of any workflow relevant data that might be used to control flow of activities as given in WFMC Interface 1 [WFMC 1].

The figure 3.1 shows the graphical representation of the process definition for above mentioned scenario. Process activities are drawn as blocks and an arrow represents transition from one activity to another along with a condition if any. The "Special Tests" activity in the diabetic patient workflow demonstrates the concept of sub-workflow and signifies how another workflow can be used as a sub-workflow to accomplish one business goal. In the figure 3.1, broken lines point to the sub-workflow "Special Tests" that further contains three activities.

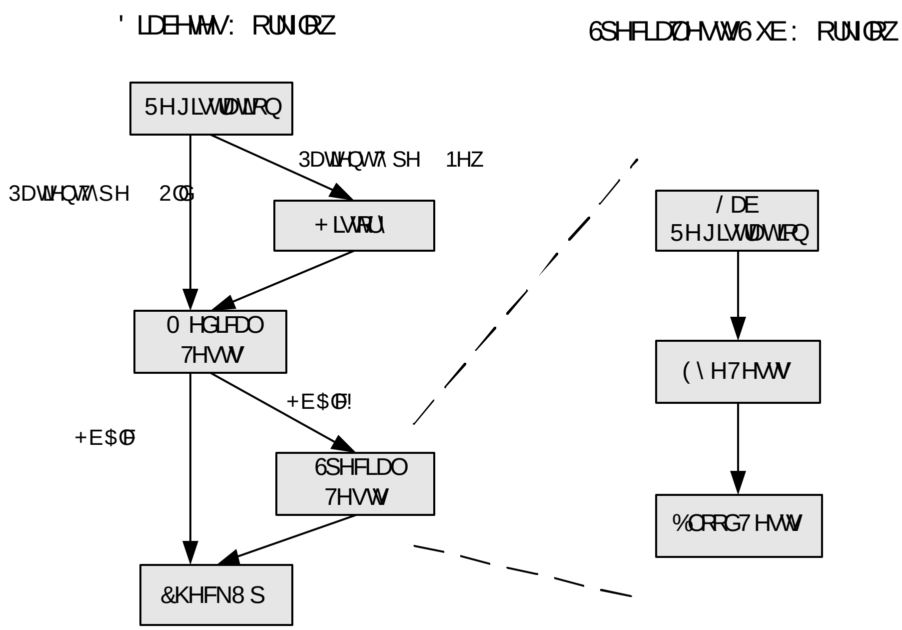


Figure 3.1: Diabetic patient workflow and special tests sub-workflow

Now we define the major constituents of the process meta model

Activity: a description of a piece of work that forms one logical step within a process.

The types of activity we support are:

- Loop activity: supports two kinds of loops, *Repeat-until* and *While-do*.

- Sub-workflow activity: this type of activity contains another workflow to have parent-child relationship.
- Routing activity: this is a dummy activity that does nothing except letting the workflow designer have more complex workflow logic.
- Atomic activity: the normal basic unit of a workflow.

Condition: a logical expression which may be evaluated by a workflow engine to decide the sequence of activity execution within a process. The relations between two conditions can be *AND* and *XOR*. The operator in a condition can be: >, >=, <=, <, !=, ==.

Restriction: The types of restrictions we support are: *XOR Split*, *AND Split*, *XOR Join* and *AND Join*. These Joins and Splits help to have a network of activities. In the given scenario, we have two Splits and two Joins and both are XOR e.g. if the patient type is “New” then the control goes to activity “History” otherwise it goes to activity “Medical Test”.

Transition: a point during the execution of a process where one activity completes and the thread of control passes to another activity.

Participant: this is a logical entity/agent/actor who is supposed to perform an activity. In process meta model the participant is an abstraction of role(s) which is defined in the role model.

Relevant data: the control data that is used by a WFMS to determine the transitions of activities.

Environment data: data that describes the computer system, on which the WFMS is running.

3.2.2 Role Model

Although the role model is not a part of WFMC specification, it is beneficial to include the role model in WFMSs to simplify the complexity of security administration. Generally security policies can be expressed in two ways: either using users or roles. The role model is used to express security policies in terms of the roles of the organization. The roles represent organizational agents to perform certain job functions within the organizations. The users in turn are assigned to the appropriate roles based on their qualifications and responsibilities. The use of roles to specify the authorizations reduces the complexity of access control because the number of roles in an organization is significantly smaller than that of the users. Moreover, the role-based authorization facilitates dynamic load balancing in workflow environments when an activity can be performed by the several individuals.

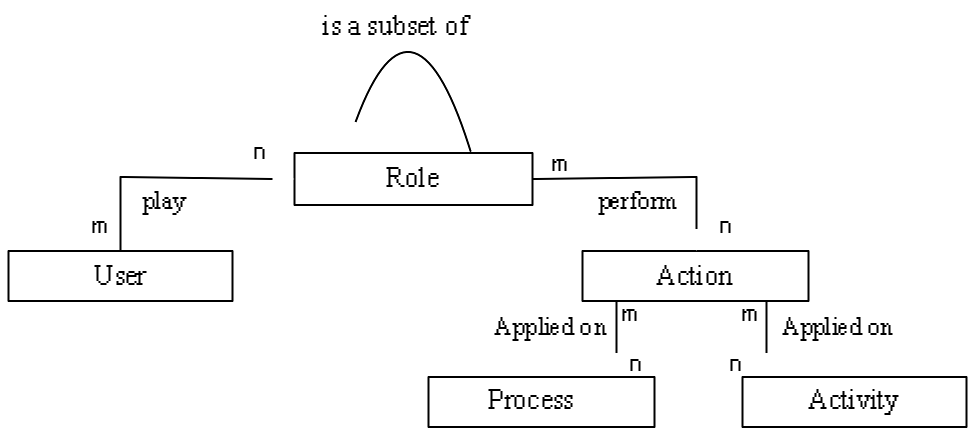
We propose a simple role model and focus on demonstrating the concept behind the role model to achieve activity assignment and access control. Following figure 3.2 shows the role meta model. The elements of the model are

Definition-Role: a *set of permissions and accessibilities e.g. a doctor or a medical nurse.*

Definition-User: a *person or an individual in the organization who can play a role, such as: Mr. Boon, Miss Lim etc.*

Definition- Action: a*n operation to be performed on an activity or a process such as create process, suspend process, execute activity or abort activity.*

By using the following simple role meta model, a tree like role hierarchy can be described. The allowed actions of the upper role in the hierarchy are the super set of all the actions that can be performed by all the lower roles.



Both process definition and role models are designed independently. The connection between role model and process definition is made through the binding of participants with particular roles. We use our example scenario to map activity role and user(s) in the diabetes department in Table1.1 and Table1.2 for diabetes and special tests workflows.

Activity	Role	User(s)
Registration	Registration Nurse	Mr. Lin, Miss Chong
History	Educator	Mr. Sam
Medical Tests	Medical Nurse	Miss Hun, Miss Lim

Check up	Doctor	Mr. Lim, Mr. Mat
Special Tests	-	-

Table 1.1: Diabetic patient workflow

Activity	Role	User(s)
Lab Registration	Laboratory Attendant	Mr. Boon
Special test	Laboratory Operator	Mr. Keat, Mr. Khim
Blood test	Laboratory Operator	Mr. Keat

Table 1.2: Special Tests sub-workflow

3.2.3 Application Model

The activities mentioned in the process definition are just logical steps. These activities have to be associated with external applications that perform the actual work. In our example, the activity "Registration" mentioned in the process definition has to be associated with a set of web pages (or web application) that get the patient's registration information and store it. In this case, we call the web application "Registration Application". We need to specify information about the web applications that are associated with the activities. As we are considering web applications, the information is simple and straight forward. This information includes:

Parameter: contains attributes to be passed into as well as out of a specific application;

Parameter-Relevant Data Mapping: Through this mapping, the system decides which relevant data is mapped to an input or output parameter of a workflow application.

Command Name: gives the location of the application. For a web application, it is application URL.

As registration activity is the first activity, it does not have input parameters but it does have output parameters (Patient ID, Patient Type). The values of these parameters are set from within the registration application. The branching condition is based on the value of Patient Type. Patient ID which is output of registration activity will be input of the next activities. The following tables show corresponding parameters and URL against each activity in our scenario. (Table 2.1, Table 2.2)

Application name	Parameters(input & output)	URL
Registration	HRN, Patient Type	http://myserver/diabetes/Start.jsp
History	HRN	http://myserver/diabetes/Educator.jsp
Medical Tests	HRN, HbAlc	http://myserver/diabetes/PatientCase.jsp
Check up	HRN	http://myserver/diabetes/Checkup.jsp
Special Tests	-	-

Table 2.1: Diabetic patient workflow applications

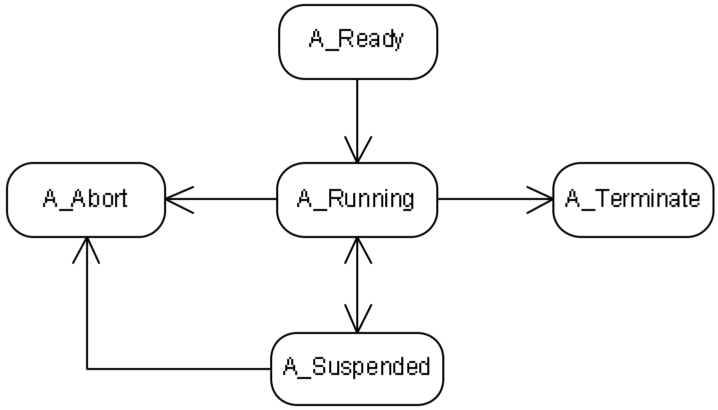
Activity	Parameters (input & output)	URL
Lab Registration	HRN	http://myserver /diabetes/Start2.jsp
Eye test	HRN	http://myserver/diabetes/EyeTest.jsp
Blood test	HRN	http://myserver/diabetes/BloodTest.jsp

Table 2.2: Special Tests sub-workflow applications

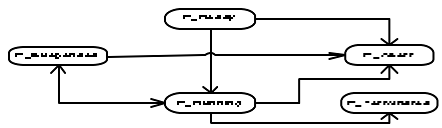
3.2.4 Activity Instance

An activity instance is a live copy of an activity defined in the process definition.

Following are the states of an activity instance in our system as shown in the figure 3.3.


A_Ready
A_Abort
A_Running
A_Terminate
A_Suspended

in the database, not as a software process. The following figure 3.4 shows the state transitions of a process instance.



3.2.5.2 Transitions

P_Ready -> *P_Running:* for this transition, the start activity of a process is instantiated i.e. the instance of the first activity is in the running state.

P_Running -> *P_Suspended:* for this transition, states of all the running activity instances are changed to the suspended state.

P_Suspended -> *P_Running:* for this transition, states of all the suspended activity instances are changed back to the running state.

P_Running -> *P_Abort:* for this transition, states of all the running and suspended activity instances are changed to the abort state.

P_Ready -> *P_Abort:* for this transition, no change in the state of the activity instance is required as there is no activity instance in the process instance.

P_Running -> *P_Terminated:* this transition occurs when the last activity instance of the process instance is terminated normally.

CHAPTER 4

META MODEL OF A VIRTUAL PROCESS

The most relevant activities within a healthcare organization can be described in the form of business processes as mentioned in [Maij 00]. Following this idea, we define a virtual process as a business process whose definition and enactment is distributed among more than one healthcare organization. From here, we define the virtual health enterprises as those whose business processes are virtual processes. The potential participating organizations do have their own internal processes which are developed previously without considering that they could be part of any virtual process. Therefore building and enacting a virtual process from such local processes is a big challenge. This challenge is intensified when dynamics of relationships, autonomous behavior of local processes and heterogeneity of WFMSs are considered. To address the challenge, changes in a process meta model and a process enactment framework are required. These changes should be such that they could easily be adopted by the WFMSs that are following the standards. In this chapter, we propose a concept of "partial views of a virtual process" to solve this problem.

4.1 OVERVIEW

To an organization a "partial view of a virtual process" is the part of whole virtual process seen by it. Such partial view of virtual process consists of organization's local process and parts of the processes of other interacting organizations. We developed a meta model for partial view of a virtual process and virtual process enactment framework. The framework is distributed and based on a collection of autonomous, physically apart, possibly heterogeneous workflow management systems connected in a loosely coupled way, most likely through the Internet. The hook-up procedure to make one virtual clinical or administrative process is bottom up. This means, firstly, the organization designs its own local process and then either exposes some of its activities for other health service providers, or connects its activities with some of the exposed activities of other healthcare service providers. Ultimately we get one virtual process, activities of which are distributed among more than one health service providers. For process modeling, we extended modeling constructs of reference meta model of process definition of the WFMC [WFMC1]. The obvious advantage of which is that it would be semantically compatible with most of the existing and coming workflow management systems. Section 2 presents meta model of partial view of virtual process and in the section 3, after presenting an illustrative example, the applications of this approach are discussed.

4.2 META MODEL OF PARTIAL VIEW OF VIRTUAL PROCESS

Broadly there have been two approaches for the design and development of the virtual processes: bottom-up and top-down. In the bottom-up approach, participating organizations design and develop their local processes independently. After having designed and developed these local processes, organizations start building up virtual process by joining their local processes with each other. In the second approach, a virtual process is first conceived, designed and then distributed among the participating organizations. The first approach has been implemented for eCommerce in a way that the local processes are wrapped up in one form or the other (mostly as services) and then they are exposed or their description is sent to one common place to form a virtual process. Our solution follows the first approach but it is different from other existing solutions. After having designed its local process, the organization defines its own partial view of the virtual process consisting of parts of external processes as well. The organization does not need to know the organizations and their processes which are beyond its view of virtual process.

Our meta model for partial view of virtual process is actually an extension of the meta model of the process definition as specified by WFMC [WFMC1]. We add five new notions which are shown as shaded blocks in the figure 4.1. For the sake of simplicity of the figure, two standard types of activity (dummy, loop) are not shown.

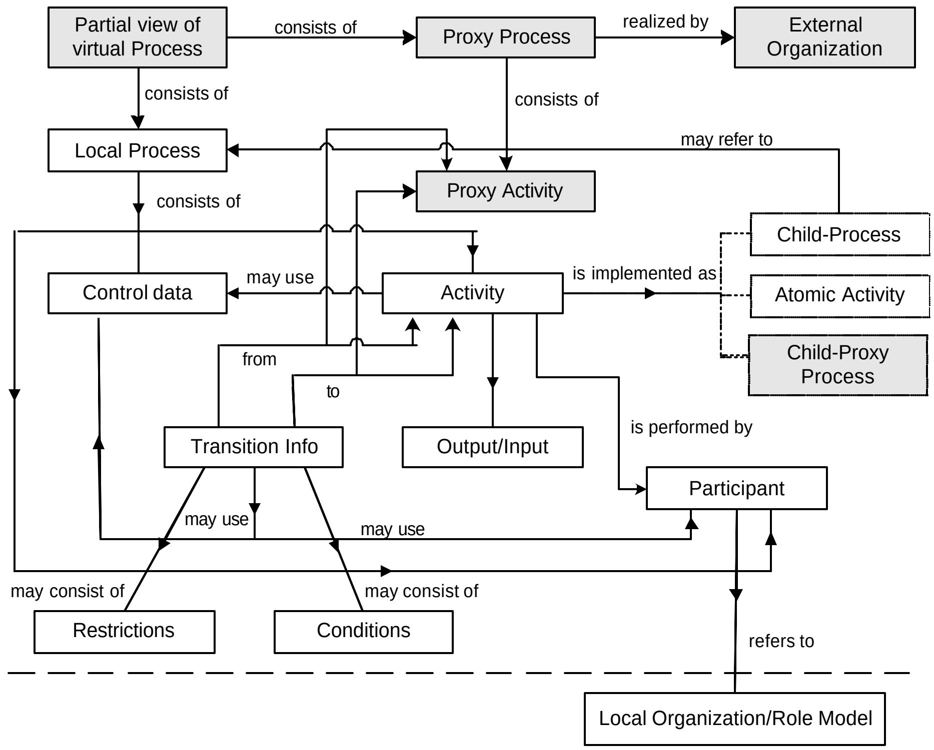


Figure 4.1: meta model of partial view of a virtual process

The entity proxy process represents an external process and during enactment it would be realized by the external WFMS. The proxy process consists of proxy activities which represent exposed activities of the external process. In other words, proxy activity or proxy process substitute some real activity or process running on some other workflow management system. From the meta model, we can define transitions from real activity to proxy activity and vice versa. However, there is no output/input and participant attached to proxy activity, as proxy activities are not supposed to produce or consume output/input. Similarly proxy activities are not meant to be executed by the local role. Entity of external organization contains the information regarding identification of external organization, communication protocol to be used, the WFMS it has and etc.

It should be noted that “proxy activity” and the activity whose type is “proxy process” are two different things. If the type of the activity is “Proxy Process” it means that the activity will act as sub-process which will be realized and executed on the workflow management system of some external organization. This is to support sub-contracting as in [15].

All the other notions in figure 4.1 are standard notions of WFMC [WFMC1]. In short, with the help of this meta model we can draw a network of processes and activities which may consist of normal processes or proxy processes. This network is the part of virtual process an organization can see or control autonomously. We name it partial view of virtual process as this is the view of virtual process seen by the organization. This meta model supports all the models specified in WFMC “interoperability models” [WFMC4].

4.3 BUILDING OF PARTIAL VIEWS OF A VIRTUAL PROCESS

4.3.1 An illustrative example

The example in figure 4.2 illustrates the application of above approach and meta model. This example is inspired by one of the scenarios given in [ERDIP], named “Acute Forensic Examination” (AFE). In figure 4.2, the local workflow processes of the participating organizations - community pediatrics, police, examination room provider and post AFE care provider – are modeled by using WFMC meta model [WFMC1]. These local workflows have been designed previously without the consideration that they would be a part of any virtual healthcare enterprise. In the figure 4.2, an oval shows one activity, which is a unit of work in a workflow process. The arrow from one oval to another symbolizes a transition from one activity to another along with the condition (the outputs of the activity are not shown here.) The splitting and joining of the activities are as mentioned in the process definition specified by WFMC [WFMC1]. For the sake of better clarity in discussion, unnecessary activities, details and issues have been omitted.

When receiving a report of alleged rape case of a child, the police assess it and then prepare a request of “Acute Forensic Examination” (AFE). The request is normally passed manually to the community pediatrics to be acted upon. While other threads of the workflow are running in parallel, this thread of activities waits for the initial AFE report to be entered. After getting the initial AFE Report and doing other activities, the police workflow waits for full AFE Report. Please note that, in this scenario, the interaction between the police and the community pediatrics is done manually. Similarly the community pediatrics has its own automated local process to entertain the request of AFE which is shown in the figure 4.2. There are two possible outcomes here: the victim is near to the community pediatrics and hence AFE is to be done locally, otherwise the AFE is to be done at a nearby hospital. For the second possibility, the request for the room is made to the nearby hospital. If AFE is performed at other hospital than both initial results and final assessment are entered manually to the local workflow of community pediatrics for the preparation of initial and final AFE reports. At the end of AFE, if necessary the patient is admitted to the post AFE care provider for further medical and psychological treatment. The local workflows of AFE room provider and post AFE care provider are self explanatory. Again only the relevant parts of processes are shown in the figure 4.2.

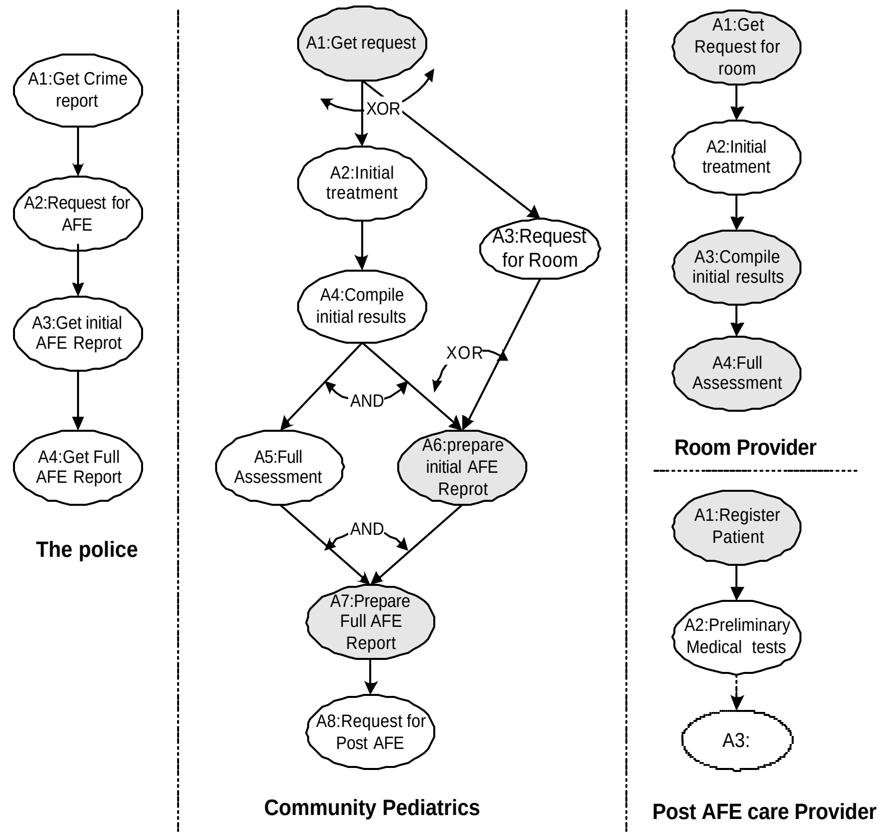


Figure 4.2: four local processes of participating organizations

4.3.2 Partial View

Suppose the four autonomous organizations now wish to establish a virtual enterprise as quickly as possible with least changes to be made in their local workflow processes created previously. This can be done in two main steps. In the first step, every organization exposes some part of the local workflow process to the designated external organization. This is achieved by creating a proxy process and its proxy activities as shown in the figure 4.3. A proxy process is created for one of the following purposes: a local workflow wants to expose some activities or a local workflow wants to access activities of other organization.

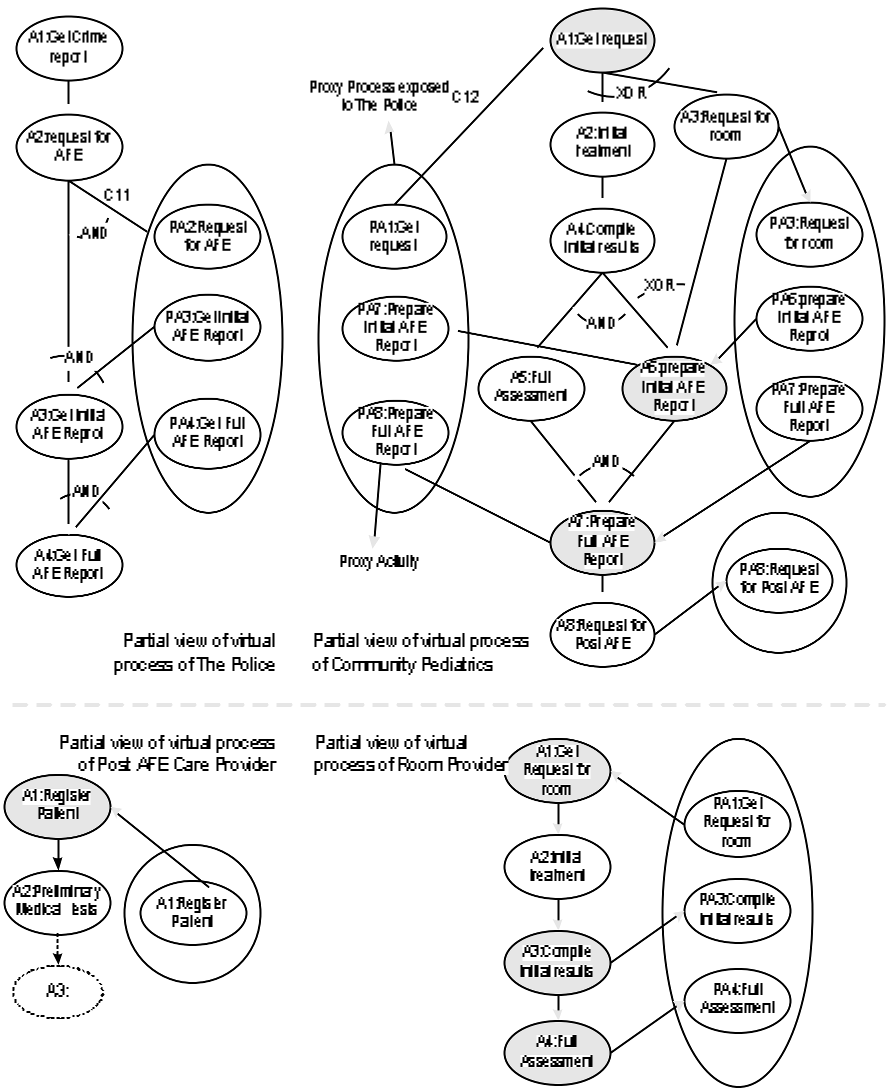


Figure 4.3: partial views of virtual processes of participating organizations

Just to explain, we focus on the police and the community pediatrics and their interactions. Based on above rules, the police create one proxy process to access community pediatrics' process. Similarly the community pediatrics creates three proxy processes: one to expose its activities to the police and two to access external processes of room provider and AFE post care provider (figure 4.3). The proxy process meant for the police has three proxy activities. These activities act as proxies of the activities A1, A7 and A8 (activities to be accessed by the police). For instance, activity PA1 is a proxy of activity A1. This proxy process is created and configured to be seen and accessed only by one organization: the police. If the same activity, say A7 of the community pediatrics, is supposed to access activity of another external organization, say room provider, then a new process activity as part of another proxy process will be created for that organization. The transitions between proxy and real activities along with the conditions are defined as part of this step. Likewise, all the organizations create necessary proxy processes and its proxy activities, and configure them to be seen by respective organizations.

The second step is to bind proxy activities of one partial view with the counter parts of other organization. For example, the police bind PA2 of its partial view with PA1 of community pediatrics. It should be noted that the workflow builder of the police see only proxy process and its three activities of the community pediatrics. All other internal details of the community pediatrics workflow are hidden from the police. After this step, the partial views of all the organizations are shown in figure 4.3.

4.3.3 Distributed Control

The transition from a real activity to a proxy activity or a proxy activity to a real activity is same as the transition between real activities. But the proxy activity itself does not do anything except two things: the communication and the distribution of control. When the transition from a local activity to a proxy activity occurs, the proxy activity sends a message to its bonded proxy activity on the other WFMS. This message contains the output of the real activity which is input of the next real activity on the receiving WFMS. This message also gives the control of the thread of activities to the workflow process on the receiving WFMS.

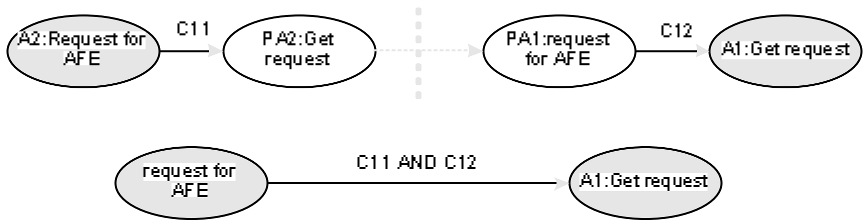


Figure 4.4: distributed control over a virtual process

Let us consider the transitions A2 ? PA2 ? PA1 ? A1 which are shown in the figure 4.4. If we remove the proxy activities, the transition becomes A2 ? A1 which is the transition of real activities belonging to two different WFMSs. For successful transition between the real activities both conditions, C11 and C12, have to be satisfied. As it is clear from the figure 4.4, the conditions are distributed and controlled independently by the two participating organizations. Hence from the perspective of the whole virtual process, the control is distributed. In such transitions, the WFMSs involved could be heterogeneous; hence there is a need to have transformation of data format and communication protocol. The proxy activities can be used to perform this necessary transformation.

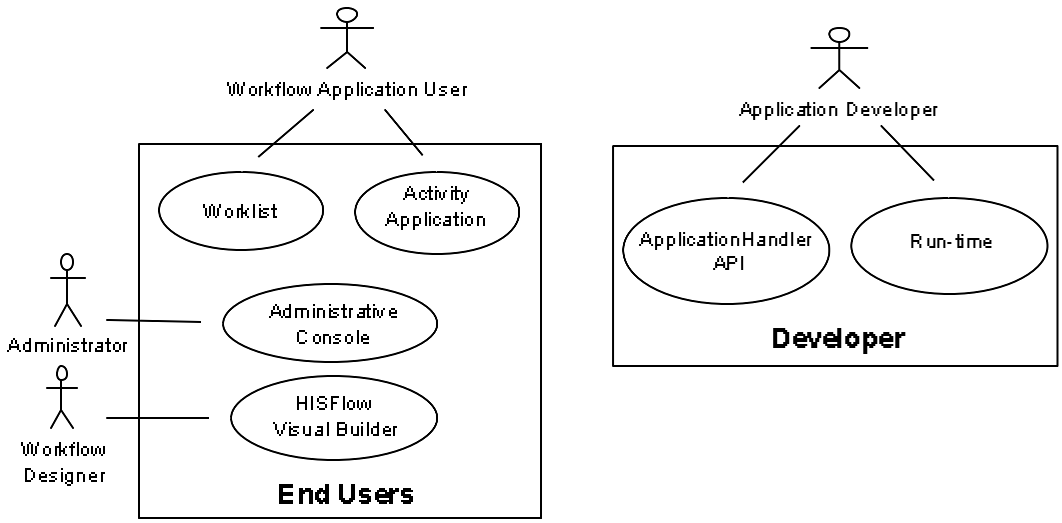
CHAPTER 5

ARCHITECTURE AND IMPLEMENTATION OF HISFLOW

After having designed a virtual process with the help of meta model of partial view of virtual process, we need a framework to enact it. This framework is called virtual process enactment framework. The virtual process enactment framework is a set of WFMS nodes that are collaborating in a very loosely coupled way. Each one of these WFMS nodes is an intra-organizational workflow management system owned by a participating organization. The set of such WFMSs collaborates with each other according to the logic defined with the help of meta model of partial view of virtual process. To understand the working of virtual process enactment, we first need to understand the architecture and working of the intra-organizational workflow management system, called HISFlow. In this chapter we present tools of HISFlow, their internal working and implementation details. In the next chapter, we present the overall architecture of virtual process enactment framework along with the models of interactions among WFMSs.

5.1 USER VIEW OF HISFLOW

From a user point of view, HISFlow can be seen as a set of tools and software artifacts with which the users interact directly. As shown in the figure 5.1, this set includes *worklist*, *administrative console*, *HISFlow visual builder*, *activity application*, *application* *handler API* and *WFMS run-time.* In Figure 5.1, each oval signifies one software artifact that is a part of HISFlow.



Workflow Application: an application that corresponds to one whole workflow and fulfils a bigger business objective i.e. a workflow application is a set of more than one activity application e.g. diabetes workflow application.

Application Handler API: an application programming interface (API) to enable the activity applications to communicate with WFMS.

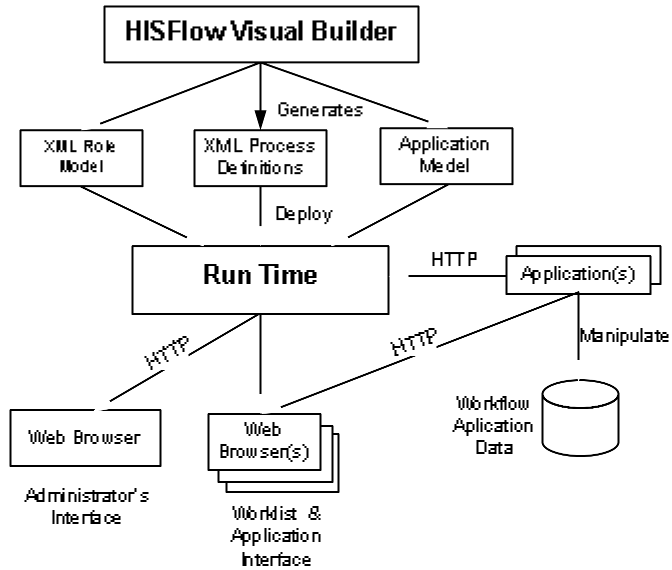
Run-time: a part of the WFMS that enacts (executes) processes.

As shown in the figure 5.1, there are two categories of the users of HISFlow: end users and application developers. The end users include workflow designer, administrator and workflow application user. Workflow designer is the one who designs and specifies workflow graphically with the help of HISFlow visual builder. Workflow designer merely specifies procedural logic or flow of activities (steps) of a workflow. These logical activities are associated with activity applications which actually perform the tasks. Application developer is the one who develops these activity applications and enables them to communicate with the WFMS by using application handler API and run-time part of the system. Here, the developers of activity applications are considered users of the system. The activity applications are meant to be used by one of the end users i.e. the workflow application user e.g. the doctors and the nurses. The workflow application user accesses the activity application (activities) through worklist.

5.2 SYSTEM ARCHITECTURE

According to [Bernstein 96], a framework is a software environment which allows it to enact applications of a specific domain effectively and efficiently. An API, a user interface and a set of tools characterize a framework. Following this definition of framework, workflow management system can be regarded as a framework.

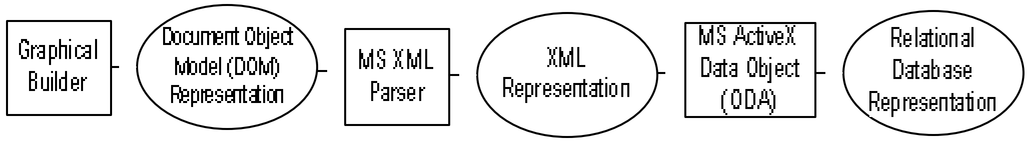
The figure 5.2 shows the sub-systems and their interaction with each other. HISFlow visual builder is a desktop application that generates three kinds of information: XML documents of the role model and the process definitions, and the application model. With the help of HISFlow visual builder, these models are deployed into run-time subsystem which is a set of components and web applications. Through the web browser, the deployed process definitions can be accessed and processes can be instantiated, monitored and controlled. Worklists are also accessible through the web browsers and when the activities are executed the corresponding applications are popped up in the same browser. These applications maintain their own data independently and WFMS does not have access to this data. However, the applications do communicate with run-time subsystem through HTTP protocol.


HISFlow Visual Builder
Generates
XML Role Model
XML Process Definitions
Application Medel
Deploy
Run Time
HTTP
Application(s)
Manipulate
HTTP
HTTP
Workflow Aplication Data
Web Browser
Administrator's Interface
Web Browser(s)
Worklist & Application Interface

that can automatically translate third party generated XML documents to our format of XML which can be deployed to our database. In short, this approach decouples the two main parts of the system, the build-time and the run-time.

The tools in HISFlow visual builder can be used independently but they are also hosted in the main host application for the sake of giving one integrated interface to the users. This brings great flexibility to the end user and facilitates the system developer to replace, modify or enhance any one of them.

The implementation of HISFlow visual builder depends largely on ActiveX components. These components decompose complex operations into small pieces and are reusable in several other tools. These modular components make the whole system easily maintainable. The platform of HISFlow visual builder is Microsoft Windows. Sybase relational database is our workflow repository. Importing XML documents of the process definitions to the database is done through Microsoft ActiveX Data Object (ADO). Visual Basic is chosen as the implementation environment for its speed and ability to build good graphical user interface. MSXML parser is used to convert DOM representation of the process definitions and the role models to XML documents, as shown in the figure 5.3.



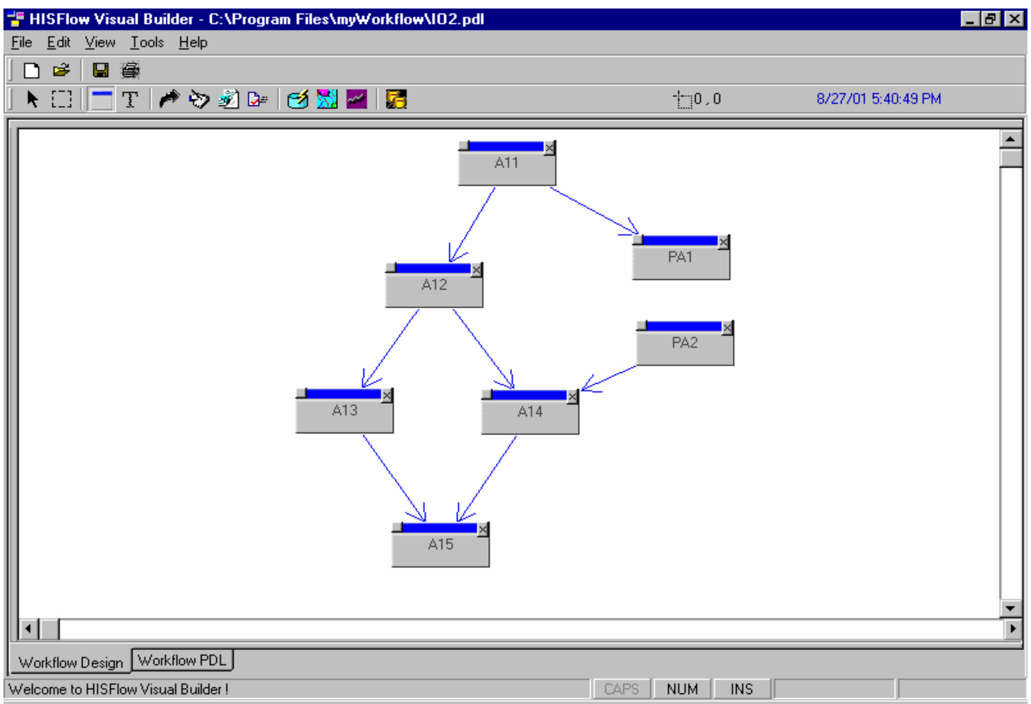


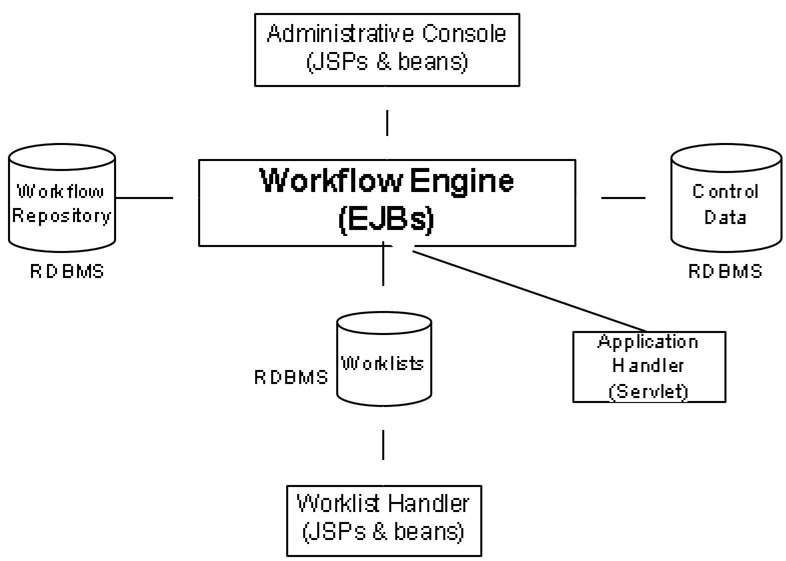
Figure 5.4: snapshot of HISFlow visual builder

5.4 RUN-TIME SUBSYSTEM

The process definitions deployed in the database are merely specifications or templates of a workflow. To execute a workflow we have to create a live copy of such process definitions, called process instances (or cases). We can have more than one instance of the same process definition out of which each and every instance has its own state. Like the process instances, we have activity instances which are live copies of the activities defined in a process definition. An activity instance to a process instance is what an activity to a process definition is.

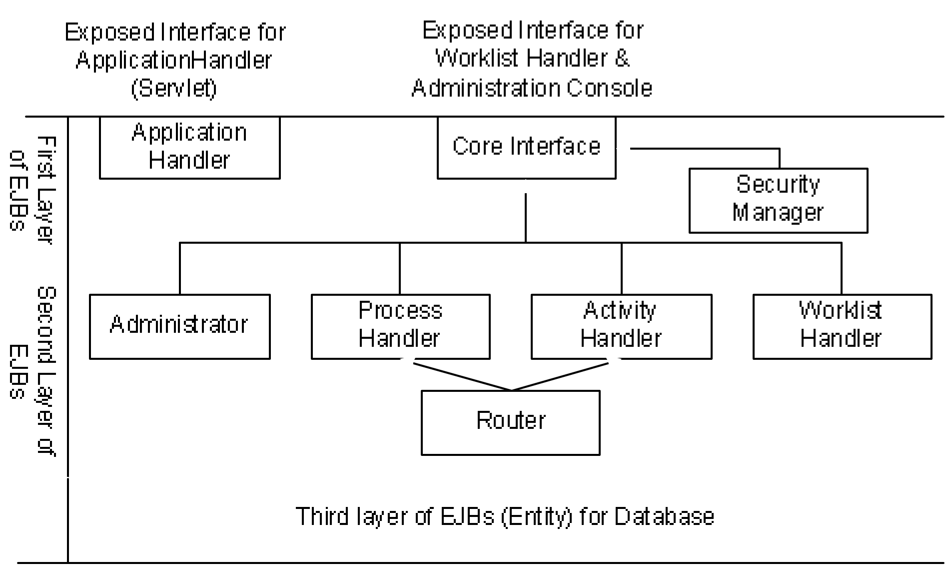
An illustrative analogy of process definitions and process instances can be taken from the object oriented paradigm. A process definition is like a class and a process instance is like an instance of the class. Moreover, a process instance is an aggregate of the activity instances. Following our example mentioned in the chapter 3 (section 3.2.1), a process instance of diabetes process definition will be created for every visit of a patient. And each process instance contains its own set of activity instances e.g. instances of registration activity, medical activity and etc.

As shown in the figure 5.4, the run-time subsystem is further divided into four functional parts: workflow engine, administrative console, application handler and worklist handler. The workflow engine is the core of the run-time subsystem that routes (schedules) the various activities by interpreting the process definitions. The administrator of the workflow management system interacts with the workflow engine through administrative console to create new process instances, and monitor and controls them. A workflow application user gets his activity instances to execute through the worklist. While the activity instances are being executed, associated activity applications are invoked. These activity applications are external to the run-time subsystem and interact with it through the application handler. There are three kinds of information stored in the runtime subsystem: workflow repository, process instances and worklists. As mentioned earlier, workflow repository is storage of database representation of workflow process definitions, which is read and interpreted by the workflow engine. The control data consists of the process instance related data including process instance states and relevant data. A worklist contains all the information about currently running activities i.e. this is the repository of all the activities to be executed in the system.



5.4.1 Workflow Engine

The workflow engine is a core of a WFMS that routes (schedules) right activities to right persons at right time by interpreting the database representation of a process definition. The workflow engine is also responsible of creating new process instances and exposes them for the purpose of monitoring. The high level design of the workflow engine is given in the figure 5.5. This three layered design of the workflow engine consists of components implemented as Enterprise Java Beans (EJBs). The layered approach is meant to divide the EJBs based on the nature of tasks they perform. The highest layer contains the components (EJBs) that mainly expose functionality of the workflow engine and implement security. The middle layer actually implements the business logic of the workflow engine. The third and the lowest layer is a set of entity beans, one of the types of EJBs, which basically facilitates other EJBs to interact with the relational database system.



5.4.1.1 Router

This component is the brain of run-time subsystem and decides what is/are the next activity/activities to be instantiated depending on two things: the logic of the process definition and the current status of the process instance. The current status of the process instance is determined by the current values of the relevant data. When an activity instance finishes, a message is passed to the Router by the CoreInterface through ActivityHandler. The Router then picks the next activities by interpreting the process definition present in the database. The activities are in turn created by the ActivityHandler component. If any of the activities happens to be a sub-workflow then an instance of the sub-workflow is created by calling a method on the ProcessHandler component.

5.4.1.2 CoreInterface

The CoreInterface is just a set of collective interfaces of all components of the middle layer. This component does not contain any business logic within it. It is primarily for two main objectives. First, objective is to give only one exposed interface to outside of the workflow engine to access all the functionalities. Second, security is enforced in this component by using SecurityManager. The obvious advantage is that the core components don’t have to bother about security issues. To show the working of CoreInterface a particular scenario is considered in the following figure 5.6.

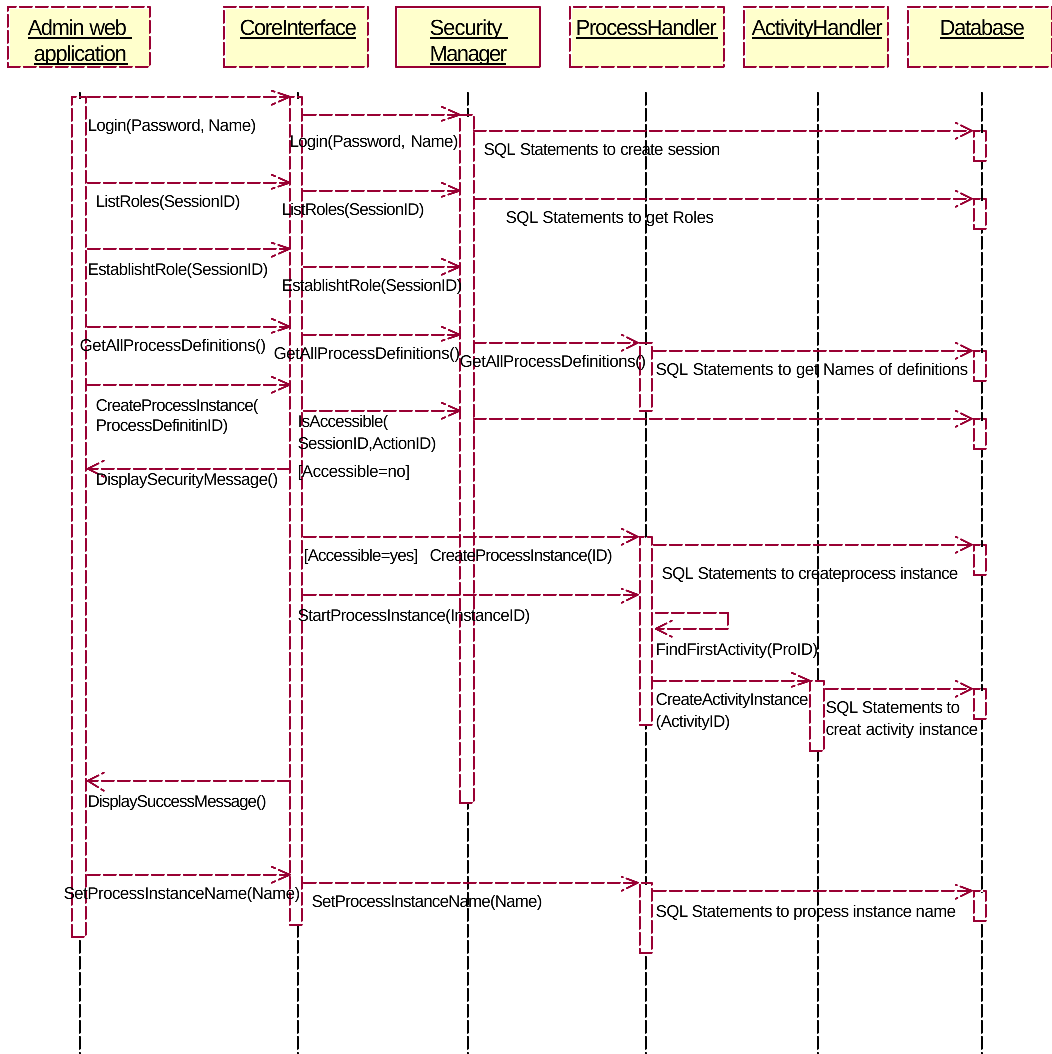


Figure 5.7: a snapshot of interaction of the components of workflow engine

5.4.1.3 SecurityManager

The current security model is simple and is based on the role model. At the design time, a workflow designer can specify the actions that a particular role can perform. When the user logs into the system as a particular role, he/she is assigned a session ID. Every subsequent request by the user is checked at CoreInterface component level against the given session ID by using SecurityManager component. The interaction among the components of the workflow engine is explained through the scenario diagram shown in the figure 5.6.

5.4.2 ApplicationHandler Servlet

There are two parts of the interaction of an external application with workflow management system: invocation of the external application and sending the data from the external application to workflow management system. The invoking of application for one particular activity instance involves getting the URL of the associated application and the parameters to be passed to the application. In addition to parameters, the context of the process instance is also passed to the application at the time of invocation as part of the URL. This context contains process instance ID, activity instance ID and the URL where the application is supposed to jump after it is done. The external application uses this context to pass the data to WFMS. For this purpose, the external application jumps to ApplicationHandler Servlet with this context as part of jump URL. ApplicationHandler Servlet passes this information to workflow engine through the exposed interface of the ApplicationHandler component. This component updates the workflow relevant data of the right process instance.

5.4.3 WorklistHandler

The worklist is a collection of activities to be performed by a particular participant (or role). Whenever any end user of that role is logged into the system, the corresponding worklist is presented to him. He can select any one of the listed activities to execute.

WorklistHandler is a set of JSPs that can be accessed through any web browser. When the user executes an activity, the activity application is popped up and when it is finished, the user jumps back to his worklist.

5.4.3 Administrative Console

The administrative console allows the user to perform administration and monitoring of the process instances. This administration and monitoring let the user create new process instances, and abort, suspend or restart the old ones. It also facilitates the user to see the details of all the previously completed or currently running process instances and their details. The administrative console is also a web application and is accessible through any web browser.

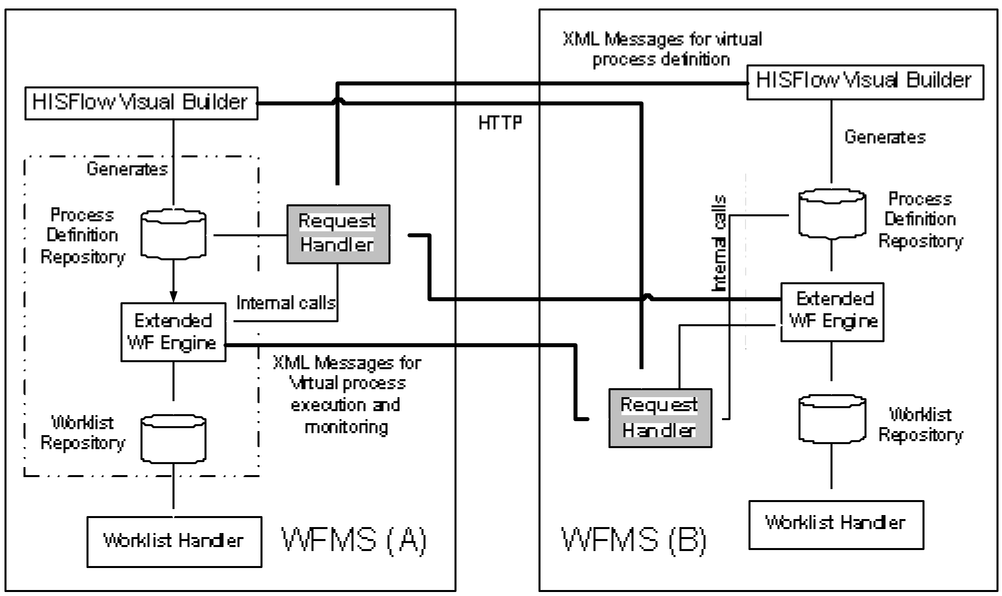
CHAPTER 6

ARCHITECTURE OF VIRTUAL PROCESS ENACTMENT FRAMEWORK

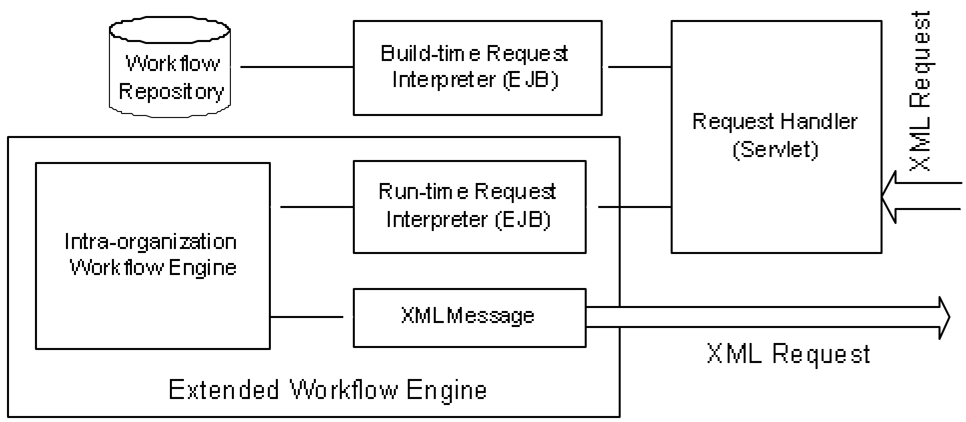
6.1 OVERVIEW

As mentioned earlier in the chapter 5, a Virtual Process Enactment Framework (VPEF) is a network of WFMS nodes. These WFMSs are maintained, monitored and controlled autonomously by the participating organizations of a virtual enterprise. Moreover, they are also likely to be heterogeneous. Hence it is required that the autonomy and heterogeneity of the WFMS are kept intact while they are participating in the VPEF. In order to enable already developed Intra-organizational WFMS take part in a VPEF, the changes should be minimized.

System architecture shown in the figure 6.1 represents major components of two WFMS nodes. Either of these nodes is linked to other through the component called *Request Handler* that acts as the only window to it. The request handler lets the participating organizations have interactions for three purposes: building, executing and monitoring of the virtual processes. At the build-time of the virtual process, HISFlow visual builder requests the request handler of other WFMS to get the exposed proxy activities. Then the HISFlow visual builder binds external proxy activities with its local proxy activities. During the execution of a virtual process instance, a workflow engine needs to interact with other workflow engine to create, synchronize and update local process instances. To accomplish all these different tasks, we need a defined set of messages to pass around among WFMS nodes. For this, we considered all three available standards: interface 4 of WFMC [WFMC4], JFlow [OMG 98] and SWAP [SWAP]. We follow the standard specified by the interface 4 of WFMC [WFMC4]. In spite of the closest match, we have to introduce new messages to cater the requirements of our approach. These required messages are mainly for viewing of exposed proxies, binding of proxies and activity-level run-time interactions.



To enable the intra-organizational WFMS to take part in the virtual process enactment framework, we need to make few changes in the workflow engine too. The extended workflow engine is shown in the following figure 6.2.



As part of either run-time interactions or build-time interactions, the intra-organizational workflow engine uses XML message component to compose the requests to be made to other WFMSs.

6.2 META MODEL OF VIRTUAL PROCESS ENACTMENT

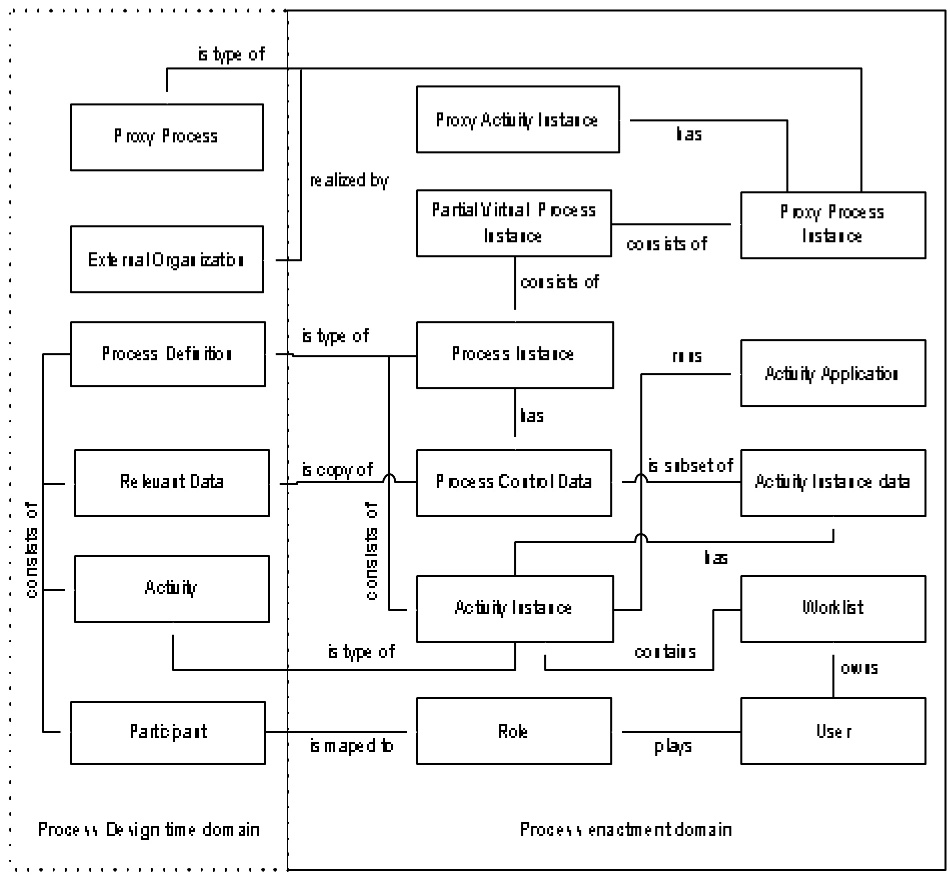
Each partial view of a virtual process is enacted independently and separately on a WFMS. But during this enactment the WFMS interacts with other WFMSs according to the definition of partial view of virtual process to achieve enactment of the whole virtual process. Like a local process definition, a virtual process may have more than one case/instance. As a virtual process definition is a set of local process definitions, a virtual process instance is a set of local process instances. Based on our meta model of partial view of a virtual process, we propose meta model of virtual process enactment. This meta model is shown in the figure 6.2 along with some entities of meta model of partial view of virtual process to show the relationship between both models.

Partial virtual process instance: it is an instance of partial view of a virtual process.

Process control data: the internal data of a process instance to control the flow of the process.

Activity instance data: a part of the process control data given to an activity.

Proxy process instance: an instance of a proxy process.



updated by the associated activity applications (or web application). An activity instance is put in to a worklist of a user who will be playing some particular role.

6.3 INTERACTION MODELS

As mentioned above, to enact a virtual process the nodes of a network of WFMS have to interact with each other. There are two types of interactions: interactions at build-time and interactions at run-time. All the interactions between two nodes of WFMSs are done through XML messages over HTTP, be it build-time interactions or run-time interactions. Interaction at build-time is done between HISFlow visual builder of the local node and the request handler of other node, while the interaction at run-time is done between the workflow engine of local node and the request handler of other node.

6.3.1 Build-time Interactions

6.3.1.1 Messages

There are two major parts of the message: the message header and the message body. The message header further consists of three elements: Request, Key and Request ID. The Request signifies that this message is a request to differentiate it from the Response Message. It also indicates if the response of the Request is required. The Key element points to the target of the request. The following is the detail of standard messages we implemented along with the little additions we made.

1. CreateProcessInstance

   This message is used to instantiate a process on other WFMS node. In this case, the Key element is the identification of process definition to be instantiated on other WFMS node. The element RequestID is the unique identification of this request. The main purpose of this unique identification is to track the requests and the responses. The following shaded block presents an example of the message to show its structure.

```
<?xml version="1.0"?>
<WfMessage Version=”1.0”>
       <WfMessageHeader>
              <Request ResponseRequired=”Yes”/>
              <Key> Process Definition ID </Key>
              <RequestID> RequestID </RequestID>
       </WfMessageHeader>
       <WfMessageBody>
              <CreateProcessInstance.Request StartImmediately=”true”>
                     <ObserverKey> Organization ID </ObserverKey>
                     <ContextData>
                     </ContextData>
              </CreateProcessInstance.Request>
       </WfMessageBody>
```

The only immediate child element of the message body represents the type of the message which is to create process instance. The child element Observer Key identifies the source of the request i.e. the identification of organization that is sending the message. The Context Data, if any, is used as initialization data of the process instance to be created.

The messages we added are

1. GetProcessDefinitions

   This message is used at build-time when the workflow designer inquires about all the process definitions of a WFMS in other organization which are public to his organization. The following example shows the structure which is self-explanatory.

```
<?xml version="1.0"?>
<WfMessage Version=”1.0”>
       <WfMessageHeader>
              <Request ResponseRequired=”Yes”/>
              <Key> </Key>
              <RequestID> RequestID </RequestID>
       </WfMessageHeader>
       <WfMessageBody>
              <GetProcessDefinitions.Request>
                     <ObserverKey> Organization ID </ObserverKey>
              </GetProcessDefinitions.Request>
       </WfMessageBody>
</WfMessage>
```

2. GetExposedActivities

   This message is a part of the build-time interactions and is used to get all the exposed activities of a particular process definition. That is why the target or Key element is an identification of process definition and the source or Observer Key is an identification of local organization.

```
<?xml version="1.0"?>
<WfMessage Version=”1.0”>
       <WfMessageHeader>
              <Request ResponseRequired=”Yes”/>
              <Key> Process Definition ID </Key>
              <RequestID> RequestID </RequestID>
       </WfMessageHeader>
       <WfMessageBody>
              <GetExposedActivities.Request>
                     <ObserverKey> Organization ID </ObserverKey>
              </GetExposedActivities.Request>
       </WfMessageBody>
</WfMessage>
```

3. BindActivities

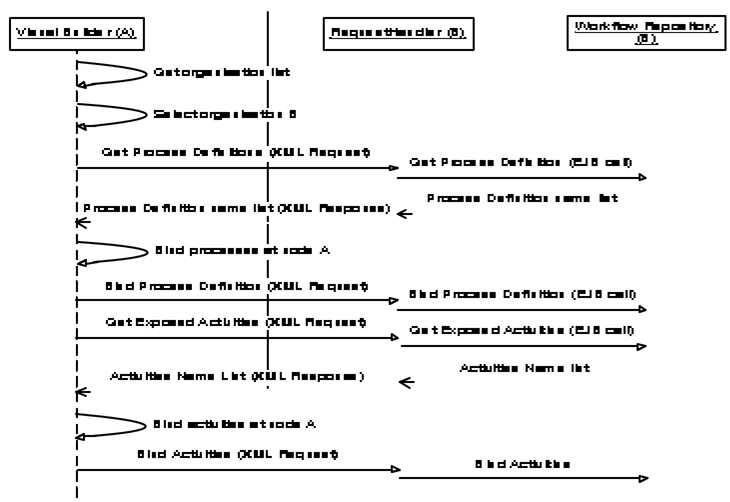
This message is used to bind the activities at the build-time. The target or Key is the external process definition i.e. the process definition on the other WFMS node. The source or Observer Key is the internal process definition. The External Activity element contains the identity of activity to which the activity of internal process definition is bonded. More than one activity can be bonded in one message.

```
<?xml version="1.0"?>
<WfMessage Version=”1.0”>
       <WfMessageHeader>
              <Request ResponseRequired=”Yes”/>
              <Key> External Process Definition ID </Key>
              <RequestID> RequestID </RequestID>
       </WfMessageHeader>
       <WfMessageBody>
              <BindActivities.Request>
                     <ObserverKey> Process Definition ID </ObserverKey>
                     <Bind>
                            <ExternalActivityID>
                                   Activity ID
                            </ExternalActivityID>
                            <InternalActivityID>
                                   Activity ID
                            </InternalActivityID>
                     </Bind>
                     <Bind>
                            <ExternalActivityID>
                                   Activity ID
                            </ExternalActivityID>
                            <InternalActivityID>
                                   Activity ID
                            </InternalActivityID>
                     </Bind>
                     ......
```

6.3.1.2 Model

The figure 6.3 shows the snapshot of build-time interactions between two WFMS nodes: WFMS (A) and WFMS (B). In the figure, the continuous horizontal lines represent request and the doted horizontal lines represent response of the request. The vertical lines signify the time. For a particular session of build-time interactions, one of the nodes acts as a server node and other acts as a client node. For the interactions shown in the figure 6.3, the node B is a server node and the node A is a client node. If we follow the build-time interactions of the police and the community pediatrics in the example given the chapter 4, the community pediatrics is a server node (B) and the police are a client node (A). The workflow designer at node (A) is building a partial view of the virtual process. The client node (A) keeps the list of addresses (URLs) of all the organizations it is supposed to interact with. The client (A) picks node (B) and makes the request to get all the process definitions which are public to node (A). It should be noted that the maintenance of names and URLs of the organizations is beyond the scope of this model and hence we suppose it is done manually. The request handler of node (B) is ready to get the request, interpret XML messages and make further requests to its workflow repository. The request handler gets the list of process definitions, compiles an XML message and sends back a response. This list is presented to the workflow designer using HISFlow visual builder of (A). The workflow designer binds the particular process definition selected from the list with the one he is currently working on. The process binding request is also sent to the server node (B) which binds the processes on its end and keeps this binding information for future interactions. Now at this point of time both process definitions are bonded to each other through their proxies. Similarly the client node (A) gets all the proxy activities of the bonded process definition, which are public to it, from the server node (B). The workflow designer picks the proxy activities one by one and binds them to the activities of local process definitions. Finally the client node (A) sends an activity binding request containing the binding information to the server node

(B). The server node (B) binds the activities on its end and saves the binding information which will be used at run-time interactions.



message is a process instance and the Key element contains identification of this process instance. In the message body, the name of the request and the state to which the process instance is moved are mentioned. The rest of the message structure is same.

```
<?xml version="1.0"?>
<WfMessage Version=”1.0”>
       <WfMessageHeader>
              <Request ResponseRequired=”Yes”/>
              <Key> Process Instance ID </Key>
              <RequestID> RequestID </RequestID>
       </WfMessageHeader>
       <WfMessageBody>
              <ChangeProcessInstanceState.Request>
                     <State>open.notrunning.suspended</State>
              </ChangeProcessInstanceState.Request>
       </WfMessageBody>
</WfMessage>
```

2. ProcessInstanceStateChanged

   This message is used to let the other WFMS node know that the state of the process instance has changed. The structure of the message is very much similar to the above messages.

4. CreateActivityInstance

   Similar to CreateProcessInstance

5. ChangeActivityInstanceState

   Similar to ChangeProcessInstanceState

6. ActivityInstanceStateChanged

   Similar to ProcessInstanceStateChanged

6.3.2.2 Model

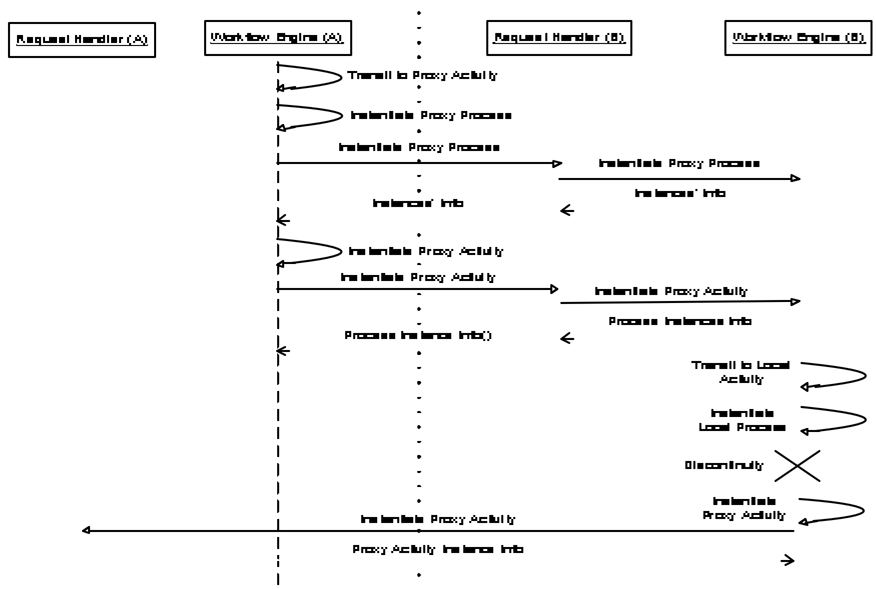
The run-time interactions occur when two process instances of the same virtual process instance running on different nodes communicate with each other. Similar to the build-time interactions, any particular session of run-time interactions is initiated by the client node (A). But, unlike the build-time interactions, the subsequent requests might be made by either of the two WFMS nodes: the client node (A) or the server node (B). There is a set of rules that governs the run-time interactions which is given below.

6.3.2.3 Rules of the game

1. If the workflow engine tries to instantiate a proxy activity which is the first in a proxy process, the proxy process is instantiated prior to the proxy activity instantiation.
2. When a proxy process is instantiated, the workflow engine makes a request to instantiate its bonded proxy process.
3. When a transition from an activity to a proxy activity is reached, the workflow engine makes a request to create bonded proxy activity of the other WFMS.
4. As proxy activity is not supposed to execute any thing, the execution of proxy activity instance means just transition to the next activity.
5. Proxy process will be instantiated if and only if the attempt is made to instantiate first proxy activity of the proxy process.

6. All the proxy activities except first will be instantiated if and only if the proxy process has been instantiated before.

A partial view of a virtual process consists of activities and proxy activities. The workflow engine has to transit from the activities to the proxy activities, while enacting the partial view of a virtual process. During this, when the workflow engine tries to instantiate the first proxy activity of the proxy process, according to the rule (1), it instantiates the proxy process first. After the instantiation of the local Proxy Process, following the rule (3), the WFMS node (A) makes the request to instantiate the bonded process which is on the other WFMS node (B). Please note that during the request and response both WFMS nodes share the identities of their newly created proxy instances with each other. These identities will be used in every subsequent call to identify the right proxy process instances. Now the first proxy activity is instantiated on the WFMS node (A). Similarly, the WFMS node (A) makes the request to instantiate the bonded proxy activity on the WFMS node (B). According to the rule (4), at the WFMS node (B), the transition from the proxy activity to the first activity of the local process will occur instantly. From this onward, the local process instance on the WFMS node (B) will start running normally. At this point of time, the initialization of the session of the run-time interactions is over. The subsequent calls of the same session can be made by either of the two WFMS nodes depending on the workflow logic designed at the build-time.


Request Handler (A)
Workflow Engine (A)
Request Handler (B)
Workflow Engine (B)
Instantiate Proxy Process
Instantiate Proxy Process
Instantiate Proxy Process
Instantiate Proxy Activity
Instantiate Proxy Activity
Instantiate Proxy Activity
Process Instance Info
Instantiate Local Process
Discontinuity
Instantiate Proxy Activity
Instantiate Proxy Activity

CHAPTER 7

WORKFLOW HISTORY MANAGEMENT

7.1 APPLICATIONS OF WORKFLOW HISTORY IN A VIRTUAL ENTERPRISE

Workflow history information includes all the history data of currently running or completed process instances. This huge amount of valuable data opens up new opportunities to further exploit workflow technology. In the literature, we have observed that the applications of workflow history information is generally scattered. Every work tries to focus on one or a few applications of workflow history information. As a part of the contribution of this thesis, we consolidate all the applications of workflow history information and discuss their relevance in a virtual enterprise. Subsequently, based on our proposed meta models, we lay down theoretical foundation for exploiting the workflow history information in the context of a virtual enterprise.

The applications of workflow history are generally divided into two broad categories: Monitoring and Controlling. Monitoring deals with the history of currently running process instances. Controlling deals with the history of already finished process instances over a longer period of time. [Muehlen, 2000] and [Muehlen, 2001] further categorize monitoring into two categories based on the purposes of the monitoring. There are two types of purposes: technical and business oriented. As the framework of a Virtual Enterprise is a set of loosely-coupled WfMS, the technical monitoring is less relevant for the virtual enterprises. For example, an organization is not concerned with the system load, response time and license management of the WfMSs of its peer organizations. But it is very much concerned to know the business states of the process instances of the peer organizations.

As one of the facets of controlling, an important application of workflow history is to do analysis of it over a very long period of time for business process re-engineering. Workflow history is analyzed to improve accuracy, efficiency and timeliness of the processes. Beate in [List, 2000] proposes a separate read-only analytical repository of history information for this purpose. This kind of analysis has new aspects in the case of Virtual Enterprise. The analysis can help the organizations to refine the current arrangements of the virtual enterprise. It can also set the guidelines for creating new virtual enterprises.

Application of the history information for sake of History-dependent Authorization [Casati, 1999] has larger scope in a Virtual Enterprise. The criteria of authorization of tasks to the users could be based on the workflow history of current instance or past instances of the peer organizations. For example, whenever user 'Y' of the peer organization executes some activity 'A', only user 'X' of this organization will execute the particular activity 'B'. Similarly workflow branching logic can also be based on the workflow history of the peer organization. In some domains like medical, workflow history serves for legal purposes as well. Future of an instance can be predicted based on the projection of the workflow history. Other applications of workflow history information include finding of workflow exception patterns to have guidelines for handling them [Sadiq, 2000] and helping the organizations maintain an Organizational Memory discussed in [Kaathoven, 1999] and [Wargetitsch, 1997]. Knowledge Management is another emerging area where benefits of workflow history are yet to be discovered fully. [Zhao, 1998] and [List, 2001] bring some of such benefits to light.

In short, not many published works talk about workflow history management. Even fewer touch this issue in the context of inter-organizational workflows or virtual enterprises. Peter in [Muth, 1999] discusses it as part of research project Mentor, but the focus is on architectural aspects of it. Querying of History Information and its optimization are the main topics of [Koksal, Mar 1998] and [Koksal, Oct 1998].

7.2 BACKGROUND

From WFMS viewpoint as explained earlier [Amin, 2002], a virtual enterprise is defined as an enterprise whose processes are virtual in the sense that they consist of geographically distributed processes. In other words, a virtual process is a set of local processes that are connected with each other to fulfill a bigger goal of the virtual enterprise. Each of such local processes runs on different autonomous WfMSs of respective organizations. As these processes are executed locally, the history information of the processes is stored in and maintained by the respective WfMSs. Such history information of the local process is only accessible to local WfMS and hence it can only be used by the respective organization. As the local process is part of the virtual process, related portion of the history of the process should also be shared with relevant organizations. Such sharing of history of local processes with the other participating organizations produces three complex problems. Firstly, we need a systematic mechanism to let the organization identify interested history information residing on other organizations. [Tagg, 2001] also raises the same problem and names it as the workflow case identifier problem. Secondly, not all the history information is meant to be shared with or accessed by other organizations and this accessibility varies from organization to organization. So we need a way to abstract history information for sake of sharing. Thirdly, we need to have an application level communication protocol for WfMSs to communicate history information among the participating organizations. In this chapter we only discuss first problem.

7.3 PROBLEM STATEMENT

A Workflow Management System of a participating organization handles its own process instances and instance related data autonomously. Hence every WfMS has its own identification of its workflow history information. Such identification is only valid inside the organization. But for different applications of history information, one organization has to refer to the workflow history information of peer organizations. Therefore, to let the organization identify workflow history information of a peer organization is the problem that has to be solved for making use of workflow history information of a virtual enterprise.

To further explain the problem, suppose two clinical processes X and Y are part of one virtual process but running on different hospitals. In the definition of X, we want to have a workflow branching condition based on the history of instances of Y. We have to put this as an expression in the definition of X so that when an instance of X is running, this expression is resolved to get the data of required instances of Y. Such expressions require two things: identification of workflow history information and some operators. The operators are beyond the scope of this thesis. We only propose the identification scheme. This scheme would provide a base not only for such expressions, but also for different kinds of analysis and monitoring.

7.4 STRUCTURE OF HISTORY INFORMATION

7.4.1 Instance Identification Scheme

Both Workflow Relevant Data and Workflow Internal Data are tied to the process instances. Once the process instance is identified, all the information attached to it can be obtained. So the first step of identification of history information is identification of process instances. Taking this observation into consideration, we define a process instance the first basic unit of workflow history information of a Virtual Enterprise. In this paper, we only propose the identification scheme of process instances across the organizations. This scheme needs to be further enhanced to have workable identification of workflow history information.

There are two levels of identification scheme: definition level and instance level. Definition level identification involves only process definitions of participating organizations. It is a step towards achieving instance level identification which involves process instances of the participating organizations. The approach of instance level identification scheme is based on the fact that a virtual process instance is a logical container that contains process instances of participating organizations. Being participants of such a logical container, process instances belonging to different organizations have same context. This same context along with definition level identification provides necessary information to identify the process instances of the peer organizations. In the following part of the paper, we develop this identification scheme by using Set theory and mathematical notations.

The participating organizations of the given Virtual Enterprise VE can be presented as

$O(VE) = \{o_1, o_2, o_3, ...o_i\}$ Where $o_1, o_2, o_3, ...o_i$ stand for the identification of organizations. And i is the total number of organizations in the Virtual Enterprise

Definition: Following the modeling approach of (Amin, 2002), Virtual Process is defined by two things: participating process definitions and peer to peer link among them. Hence, Definition of Virtual Process VP can be represented as

$VP = \{P(VP), \|(VP)\}$ Where $P(VP)$ means all the process definitions participating in Virtual Process Definition VP and $\|(VP)$ means all the peer to peer relations of the participating process definitions of VP.

All the process definitions belonging to organization O_1 are represented as

$P(O_1) = \{p_1, p_2, p_3, ...p_{k_1}\}$ Where $p_1, p_2, p_3, ...p_{k_1}$ are identifiers of process definitions of organization O_1 which are unique within the organization. And k_1 is the total number of process definitions in organization O_1.

Please note that $P(O_1)$ should be read as P of O_1 meaning the processes of O_1.

Similarly for organization O_i, its process definitions can be represented as

$P(O_i) = \{p_1, p_2, p_3 ..... p_{k_i}\}$ Where $p_1, p_2, p_3 ..... p_{k_i}$ are identifiers of process definitions of organization O_i which are unique within the organization. And k_i is the total number of process definitions in organization O_i.

The participating organizations of Virtual Process Definition VP can be represented as

$O(VP) = \{o_1, o_2, o_3, ... o_j\}$ Where $O(VP) \subseteq O(VE)$ and j is the total number of organizations participating in Virtual Process Definition VP.

A process definition can uniquely and globally be identified with the combination of two identifiers: the locally unique process definition identifier and its organization identifier, provided the organization identifier is globally unique. For example process definition $p_{(a,b)}$ is globally unique, where 'a' stands for process definition identifier and 'b' stands for organization identifier. For sake of simplicity we assume that only one process definition of the organization participates in the Virtual Process Definition VP. In terms of globally unique identifiers, the participating process definitions in Virtual Process Definition VP can be represented as

$$P(VP) = \{p_{(m_1,o_1)}, p_{(m_2,o_2)}, p_{(m_3,o_3)}, ... p_{(m_j,o_j)}\}$$

Where $m_1, m_2, m_3, ... m_j \in P(O_1), P(O_2), P(O_3), ... P(O_j)$ respectively and $o_1, o_2, o_3, ... o_j \in O(VP)$

For nth Virtual Process Definition

$P(VP_n) = \{p_{(m_j^n, o_j)}\}$ …………….. (A)

We represent peer to peer relations of processes definition p_1 with other process definitions, say p_2 and p_3, as $p_1 \| (p_2, p_3)$.

Therefore all the peer to peer relations of $P(VP)$ in the Virtual Process Definition VP can be represented as

$$\|(VP) = \{p_{(m_1,o_1)} \| (p_{(r_1,q_1)}), p_{(m_2,o_2)} \| (p_{(r_2,q_2)}), \ldots\ldots p_{(m_j,o_j)} \| (p_{(r_j,q_j)})\} \ldots\ldots \text{(B)}$$

Where $q_j \subseteq O(VP)$ and $m_j \notin r_j$ and

$$(r_j, q_j) = \{(\Phi_1, l_1), (\Phi_2, l_2), (\Phi_3, l_3), \ldots\ldots\ldots (\Phi_h, l_h)\}$$

Where $h = card(q_j)$ and $l_h \in q_j$ and $\Phi_h \in P(l_h)$

and $l_a \neq l_b$ if $a = b$

After having expressions (A) and (B), we apply above identification scheme on the Partial View concept of (Amin, 2002).

Definition: As defined in (Amin, 2002), a Virtual Process Definition is a set of Partial Views seen by individual participating organizations.

$VP = \{PV_1VP, PV_2VP, PV_3VP, \ldots\ldots\ldots PV_jVP\}$ Where PV_1VP, PV_2VP, PV_3VP and PV_jVP are Partial Views of VP seen by the organizations o_1, o_2, o_3 and o_j respectively.

Definition: To an organization, Partial View of a Virtual Process Definition is a set of its local process definition and peer to peer relations with other participating process definitions of the same Virtual Process Definition.

Following above definition of Partial View, PV_1VP can be represented as

$$PV_1VP = \{(p_{(m_1,o_1)}),(p_{(m_1,o_1)} \| p_{(r_1,q_1)})\}$$

Similarly

$$PV_jVP = \{(p_{(m_j,o_j)}),(p_{(m_j,o_j)} \| p_{(r_j,q_j)})\}$$

For jth Partial View of nth VP

$$PV_jVP_n = \left\{\left(p_{(m_j^n,j)}\right),\left(p_{(m_j^n,j)} \| p_{(r_j^n,q_j^n)}\right)\right\} \text{............ (C)}$$

With the help of above expression definition of local process and its relations can be identified. Please note that Proxy Process of Meta Model of Partial View given in (Amin, 2002) is just a way of implementation of relation of local process with other processes.

As mentioned above, notion of Virtual Process Instance provides a context that logically connects all the participating process instances belonging to different organizations. All the instances of Virtual Processes Definition VP can be represented as

$I(VP) = \{I_1VP, I_2VP, I_3VP, ...I_gVP\}$ Where I_gVP is the identifier of gth instance of Virtual Process Definition VP.

Following the expression (A)

$I_gVP = \{I_g\,p_{(m_j,o_j)}\}$ Where $I_g\,p_{(m_j,o_j)}$ are all the process instances of participating organizations that belong to Virtual Process Instance I_gVP.

Similarly, for the gth instance of Partial View PV_jVP of Virtual Process Definition VP can be represented as

$$I_g PV_j VP = \left\{ I_g\, p_{(m_j, j)}, I_g \left(p_{(m_j, j)} \middle\| p_{(r_j, q_j)} \right) \right\}$$ ………. (D)

Where $I_g \left(p_{(m_j, j)} \middle\| p_{(r_j, q_j)} \right)$ is the g*th* instance of relations between local process definition $p_{(m_j, j)}$ and $p_{(r_j, q_j)}$. Remember $p_{(r_j, q_j)}$ are process definitions of peer organizations with which local process definition is connected.

The semantic of instance of relation, say $I_a \left(p_b \| p_c \right)$, is one complete interaction between two process instances $I_a p_b$ and $I_a p_c$. Where $I_a p_b$ and $I_a p_c$ are a*th* instances of process definitions p_b and p_c respectively. Process instance $I_a p_b$ is a local process instance and $I_a p_c$ is a process instance of a peer organization. The expression (D) let the organization know which process instances of peer organizations are interacting with which local process instances. In simpler words, the relationship of local process instances and process instances of peer organizations is captured in expression (D). And this relationship helps the organization identify process instances of peer organizations.

7.4.2 Workflow History Space

7.4.2.1 Process/Place/Time Model

We conceptualize Workflow History Information as three co-ordinate space, axes of which are Process, Place and Time. "Process" and "Place" symbolize Virtual Process Definition and Organization respectively. A single point in this space is a set of instances of a process definition belonging to an organization. For example, a point $\left(VP_1, O_2, T_6 \right)$ shown in "Figure 1 (a)" gives us a set of all instance identifiers of a local process definition $p_{(m_2^1,O_2)}$ that exist at time T_6. And as we know from expression (A), $p_{(m_2^1,O_2)}$ is a participating process definition of VP_1 and belongs to the organization O_2. This set of instances includes both the instances that are currently active and the instances that have been executed by the time T_6. In this view of the workflow history information space, we only consider to identify the instances, not the internal details of the instances.

7.4.2.2 Instance/Place/Time Model

This model gives another view of the workflow history that can be used to have internal detail of a process instance. Axes of this space are Instance, Place and Time. The "Instance" symbolizes a Virtual Process Instance. A single point in this space is a snap shot of a local process instance belonging to an organization at given time. For example, a point (I_3VP_1, O_1, T_4) shown in "Figure 1 (b)" gives a snap shot of a local process instance $I_3 p_{(m_1^1,O_1)}$ of organization O_1 at time T_4 that belongs to I_3VP_1. By snap shot we mean the values of all kind of data attached to the process instance at given point of time. This includes the data and the states of both finished and active activities, the relevant data, and the instance internal run-time data.

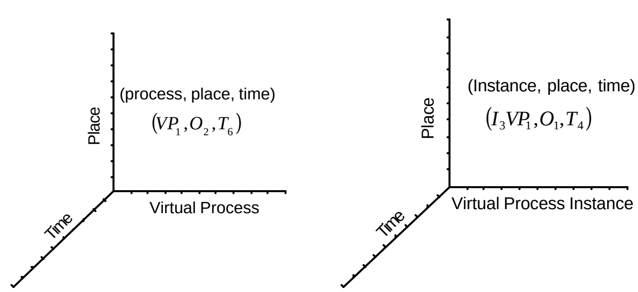


Figure 1 – (a) Virtual Process Space (b) Virtual Process Instance Space

7.5 APPLICATION OF INSTANCE IDENTIFICATION SCHEME

A simplified form of the example given in the chapter 5 is used to demonstrate the application of proposed instance identification scheme. We ignore the internal details of the process definitions as we are not concerned with it. We also add few extra process definitions into picture to make it more suitable for the current purpose. Four organizations, the Police, the Community Pediatrics, the Examination Room Provider and the Post AFE Care Provider, are collaborating to form a Virtual Enterprise. A child rape case is reported to the Police and it asks the Community Pediatrics to do AFE examination. The Community Pediatrics requires an examination room from nearby hospital, the Examination Room Provider. After the initial AFE examination, the patient is admitted to the Post AFE Care Provider for further treatment. All the four participating organizations have their own processes that interact with each other to form a Virtual Process to achieve the bigger goal.

As shown in the “Figure 2”, the process definition P2 of The Police is participating in the Virtual Process Definition VP_1, and linked with one of the process definition P3 of The Community Pediatrics. Similarly process definition P3 of Community Pediatrics is further interacting with process definitions P1 and P4 of Post AFE Provider and Room Provider respectively.

Suppose VP_1 denotes the virtual process given in the example. By using equation (A), all the participating process definitions of VP_1 are presented in terms of globally unique identifiers as

$$P(VP_1) = \{p_{(p_2,o_1)}, p_{(p_4,o_2)}, p_{(p_1,o_3)}, p_{(p_3,o_4)}\}$$

By using equation (B), all the relations among the local process definitions of VP_1 are presented as

$$\|(VP_1) = \begin{Bmatrix} p_{(p_2,o_1)} \| (p_{(p_4,o_2)}), p_{(p_4,o_2)} \| (p_{(p_2,o_1)}, p_{(p_1,o_3)}, p_{(p_3,o_4)}), \\ p_{(p_1,o_3)} \| (p_{(p_4,o_2)}), p_{(p_3,o_4)} \| (p_{(p_4,o_2)}) \end{Bmatrix}$$

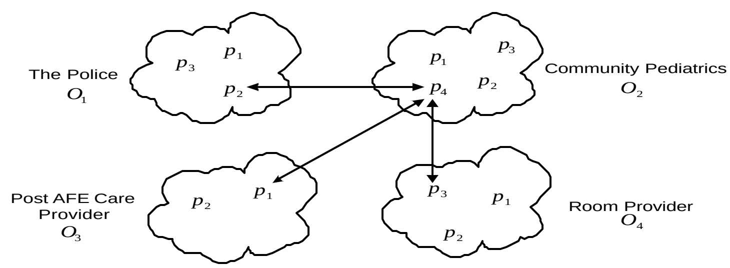


Figure 2 – Interaction of participating processes of a Virtual Process

For the Partial View of VP_1 seen by the Police in terms of global identifiers can be represented as

$$PV_1VP_1 = \{(p_{(p_2,o_1)}), (p_{(p_2,o_1)} \| (p_{(p_4,o_2)}))\}$$

Similarly for the Partial View seen by Community Pediatrics

$$PV_2VP_1 = \{(p_{(p_4,o_2)}), (p_{(p_4,o_2)} \| (p_{(p_2,o_1)}, p_{(p_1,o_3)}, p_{(p_3,o_4)}))\}$$

For the Partial View seen by Post AFE Care Provider

$$PV_3VP_1 = \{(p_{(p_1,o_3)}), (p_{(p_1,o_3)} \| (p_{(p_4,o_2)}))\}$$

For the Partial View seen by Room Provider

$$PV_4VP_1 = \{(p_{(p_3,o_4)}), (p_{(p_3,o_4)} \| (p_{(p_4,o_2)}))\}$$

Now let us take an example of one instance of the partial view seen by Community Pediatrics, say $I_3PV_2VP_1$. Following expression (D), Community Pediatrics can identify process instance $I_3 p_{(p_2,o_1)}$ belonging to the police. Similarly other process instances can also be identified.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

In order to be competitive, organizations need to deploy information technology solutions that internally automate the paper-based record systems and externally create smart connections between the major participants. With the emergence of near ubiquitous Internet connectivity, a virtually integrated enterprise built upon connections, partnerships, alliances, and relationships with partners is arising as the operating model for the organizations. This new business model leads to the concept of Virtual Enterprises (VE) that is the foundation of the networked economy. The original goals for virtual enterprise business systems are to enable deployment of distributed business processes among different partners, to increase the efficiency of existing provided services, to decrease the cost for the provision of these services, and to adapt rapidly to new market changes.

Although from business point of view virtual enterprises are the most promising business model, from technical point of view, the required solutions and systems are more complex, sophisticated and distributed [Ducroux, 1998]. Current technologies and scientific results are not addressing in a consistent and coherent way certain key requirements of the virtual enterprises. Open issues like specification of business processes in the context of virtual enterprise, flexible and dynamic mechanisms for autonomous, distributed, and loosely coupled execution of business processes across organizational boundaries are not addressed. As many organizations have adopted intra-organizational WFMSs, the integration of existing intra-organizational WFMSs with each others to form the infrastructure of virtual enterprise is highly desirable. Additionally, the deployment of tightly coupled communication mechanisms, like CORBA, is rather problematic and in general has made current solutions inflexible. Therefore, it is anticipated that a message-based approach with corresponding XML message requests and responses would have been better since the degree of autonomy and flexibility is increased [Georgakopoulos 98, Miller 98].

Our project HISFlow is an attempt to contribute towards simplifying intra-organizational workflow management foundation by harnessing new technologies. As a part of this thesis, we present architecture and implementation of HISFlow that consists of two sub-systems: a graphical build-time subsystem and a run-time subsystem. The workflow engine, the core of run-time, is implemented in J2EE and is capable to be integrated with web applications. An illustrative workflow application for managing diabetes patients in a hospital is also presented. As defined in [Miller 97], workflow management systems that provide web interfaces are called web-enabled, while if web technologies are the only infrastructure used to build the workflow management system, providing both interfaces and communication/distribution, are called web-based. Most of the available intra-organizational workflow management systems are merely web-enabled, not web-based [Miller 97]. In contrast, HISFlow gets an edge by exploiting technologies like EJB, XML and JSP to implement fully web-based workflow architecture.

According to the categorization of [Miller 96], workflow engine of HISFlow is centralized i.e. the decisions like which activity/activities should be performed when and by who are made centrally. It means Control dimension of the HISFlow is centralized. As external applications are web based and may reside anywhere on the World Wide Web along with their data, the Application Data and Execution Script (Applications) dimensions are dispersed.

In a virtual enterprise, the potential participating organizations do have their own internal processes which are likely to be developed without considering that they could be part of any virtual enterprise. We propose a solution for building and enacting a virtual process from such internal processes. For this, we introduce a concept of "partial view" which is, to an organization, a part of whole virtual process seen by it. We develop meta models both for such partial views of virtual processes and entities of virtual process enactment. The framework to enact virtual processes is distributed and based on a collection of autonomous, physically apart intra-organizational WFMSs that are connected in a loosely coupled way through the Internet. Based on the meta-models, we also demonstrate the design and architecture to convert an intra-organizational WFMS to the one which can take part in a virtual enterprise. For this, we develop interaction models to let the participating WFMSs communicate with each other. We use the interoperability characteristic of XML to implement the interaction models by defining a set XML messages.

There are various applications of workflow history information maintained by Workflow Management System in the organizations. Most of such applications are also relevant and required in the context of a virtual enterprise. To make these applications of workflow history information feasible, sharing of geographically distributed history information among participating organizations of a virtual enterprise is vital. Such sharing requires a common systematic way of identification of history information. As a part of development of HISFlow, we develop a simple and generic scheme to identify history information of workflows across a virtual enterprise. This scheme lays down a theoretical foundation to be used for workflow history management of a virtual enterprise. The scheme also caters the concept of partial view of a virtual process. We use an example to demonstrate the working of the scheme for the identification of workflow history information.

There are several advantages of our approach of building virtual processes which make our system more practical. The concept of partial views lets the organizations make their existing workflows be able to create a virtual process quickly and it could be achieved with the least changes in their existing workflows. This allows the participating organization to design and implement their local workflows independently.

The second advantage is that meta model of partial view is based on international standard of process definition specified by WFMC [WFMC1]. For being based on this standard we believe most of the existing and coming workflow management systems would be able to adopt our solution. The third and what we believe the most important gain is dynamic and elastic nature of a virtual process. By dynamicity we mean the relationships of participating organizations can be changed quickly. Elasticity signifies that a virtual process can be extended as much as it is required. In the example given in the chapter 4, suppose, a medical insurance company of the patient wants to be part of the virtual enterprise and to link up its payment workflow process with Post AFE Care Provider. For this, what it needs to do is define its own view of virtual process and link it with partial view of virtual process of Post AFE Care Provider the way it is done in the example. Clearly, in this scenario, it is fast and transparent to other organizations which are the Community Pediatrics, the Police and the Room Provider.

Our solution also offers generality and applicability in various applications areas and business sectors due to the generality of the business process definition language, although we pick healthcare sector to demonstrate the ideas.

For workflow history management, although we refer to some particular process model, in the thesis we try to keep our discussion and solution at abstract level and independent of details of any process model. This makes the approach equally useful for any process model. But on the other hand, because of being primitive, the solution is not complete enough to be practical unless it is enhanced to cover detailed data of process instances The main focus is to solve the problem of identification and linking of the processes of a virtual enterprise for sake of history information. We believe that the approach can be useful in any form of process automation that involves geographically distributed processes.

A shortcoming of our solution is the difficulty to ensure the correctness of a virtual process. A virtual process consisting of correct local workflows could be incorrect due to the incorrect connections among the local processes. The existing process building and analysis tools could be unable to make sure that the overall virtual process developed by our approach is correct. To overcome this problem, researchers may have to introduce new techniques and tools. In spite of this, we still believe in case of simple interactions among participating organizations this shortcoming is not significant enough if we consider all the gains.

8.2 FUTURE WORK

Workflow History Information: As mentioned earlier, we only proposed theoretical foundation for workflow history information. We did not consider internal details of process instances. To exploit the approach fully, more work needs to be done to include all kinds of data related to process instances. Design and implementation of a complete workflow history management solution should follow the theoretical work.

Enhanced Role Model: the role model currently used is primitive; it can not support modeling of complex organizational hierarchy. A more comprehensive role model is presently under development by other members of our group. This new role model supports hierarchical roles, use of constraints to support dynamic separation of duties and assignment of activities, and use of community to define a context for a group of people belonging to different roles. The role model also offers the use of role stages to define the qualification level for different users, dynamic load balancing of activities between users, and inter-organizational access control, which allows an organization to grant access privileges to their resources to other organizations based on their business agreements.

Correctness and analyzing tool: as a virtual process consisting of correct local workflows could be incorrect due to the incorrect connections among the local processes, a tool needs to be developed to ensure the correctness of the virtual processes and to perform their analyses.

Exception handling: a process model can not capture all the possibilities in a workflow at build-time. Any abnormal situation during run-time may cause a workflow exception. As a future work, a complete exception handling mechanism should be implemented.

Process definition using WFMC standard XPDL: as WFMC has issued an XML Process Definition Language (XPDL) in May 2001, we can use this standard XPDL, instead of proprietary XML schema in our HISFlow visual builder to specify the process definitions. This work can be done by rewriting the "XML generation" part of the HISFlow visual builder and the other parts of it will not be affected.

REFERENCES

[Aalst 01] Aalst, W.M.P., M. Weske, The P2P Approach to Interorganizational Workflows, Proceedings of the 13th International Conference on Advanced Information Systems Engineering (CAiSE'01), pages 140-156 Springer-Verlag, Berlin, 2001

[Aalst 99] Aalst, W.M.P., Anyanwu, K.: Inheritance of Interorganizational Workflows to Enable Business-to-Business E-commerce, Proceedings of the Second International Conference on Telecommunications and Electronic Commerce (ICTEC'99), pages 141-157, October 1999

[Abbot 94] Abbot, K. R. and Sarin, S. K., "Experiences with Workflow Management: Issues for The Next Generation," Computer-Supported Cooperative Work (CSCW 94), 1994.

[Adams 97] Adams, T. Dworkin 1997 "Workflow Interoperability Between Businesses". In: Lawerence P. (ed.) WfMC Workflow Handbook. John Wiley & Sons, New York, 1997, pp 211—221.

[Ader 97] Ader, M. "Seven Workflow Engines Reviewed", Document World Vol. 2, No. 3 May/June 1997

[AdDadam 00] [2] AdDadam, P.; Reichert, M.; Kuhn, K.: Clinical Workflows - The Killer Application for Process-oriented Information Systems? Proc. 4th Int'l Conf. on Business Information Systems (BIS'2000), Poznan, April 2000, pp. 36-59

[Alonso 99] Alonso, G. Fiedler, U. Lazcano, A. Schuldt, H. Schuler, C. Weiler N. (1999) "WISE: An Infrastructure for E-Commerce." Proceedings of the Informatik'99 Workshop "Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications. Paderborn, Germany, October 6, 1999.

[Amin 02] 1. Amin T., Keng P. H.: Inter-organizational Workflow Management System for Virtual Healthcare Enterprise (To be appear), 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises PRO-VE'02, Portugal, May 1-3, 2002.

[Applegate 96] Applegate, L M, Holsapple C W, Kalakota R, Radermacher F J and Whinston A B (1996) Electronic Commerce: Building Blocks of New Business Opportunity, Journal of Organizational Computing and Electronic Commerce vol. 6, no. 1. pp. 1-10.

[Banahan 99] Banahan, E., Banti, M. (1999), Report on Workshop on Legal Aspects of Virtual Organisations, Brussels, 30 November 1999, http://www.ispo.cec.be/serist/

[Bernstein 96] [11] Bernstein, P.: Middleware: A Model for Distributed System Services. Communications of the ACM, Vol. 39, No. 2, February 1996, pp. 86-98

[Billington 94] Billington C (1994) "Strategic Supply Chain Management" OR/MS Today, April 1994. pp. 20-27.

[Bolcer 99] Bolcer, G. A. Kaiser, G. (1999) "SWAP: Leveraging the Web to Manage Workflow" IEEE Internet Computing Jan/Feb 1999

[BPMI] [6] The Business Process Management Initiative, http://www.bpmi.org/specifications.esp

[Camarinha 99a] Camarinha-Matos, L. M., Afsarmanesh, H., (eds) (1999a), Infrastructures for Virtual Enterprises. Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, 27-28 October 1999, Boston: Kluwer Academic Publishers

[Camarinha 99b] Camarinha-Matos, L.M., Afsarmanesh, H. (1999b), "The PRODNET Demonstrator" in Infrastructures for Virtual Enterprises. Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, 27-28 October 1999, Boston: pp. 279-290

[Camarinha 99c] Camarinha-Matos, L.M., Afsarmanesh, H. (1999c), "The Virtual Enterprise Concept" in Infrastructures for Virtual Enterprises. Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, 27-28 October 1999, Boston: pp. 3-30.

[Carr 96] Carr J (1996) Intranets Deliver. InfoWorld vol. 18, no. 8. pp. 61-63.

[Casati 99] 2. Casati, F., Castano, S., Fugini, M. G.: Managing Workflow Authorization Constraints Through Active Database Technology", Information Systems Frontiers, 3(3). 1999.

[Ciara 99] Ciara Byrne (1999) "FACTS and Fiction: Agents in the Electronic Travel Market", Communicate, Vol 4, issue 2,

[Deloitte 00] Deloitte, Touche: The emerging European Health Telematics Industry, Reference C13.25533, February 2000

[Diabetes] [5] http://www.drugdispensary.com/healthcare/diabetes.html

[Doz 98] Doz, Y.L., Hamel, G. (1998), "Alliance Advantage. The Art of Creating Value through Partnering", Boston: Harvard Business School Press.

[Ellis 00] [13] Ellis, C.: A Framework and Taxonomy for Workflow Architecture. In: Dieng, R., Giboin, A., Karsenty, L., De Michelis, G. (eds.): Designing Cooperative Systems – The Use of Theories and Models. IOS Press-Ohmsha (2000) 99-112

[ERDIP] ERDIP, Cross Boundary Information Flows , http://www.nhsia.nhs.uk/erdip/pages/publications/DemonCrossBoundaryInformationFlows.pdf

[Fielding 98] Fielding R. (1998) "Support for the Virtual Enterprise: web-based Development of Complex Information Products" Communications of the ACM, Vol. 41, No 8, August 1998, pp. 84-92

[Filos 00] Filos, E., Ouzounis, V. (2000) "Virtual Organisations: Technologies, Trends, Standards and the Contribution of the European RTD Programmes" International Journal of Computer Applications in Technology, Special Issue: "Applications in Industry of Product and Process Modeling Using Standards"

[Georgakopoulos 99] Georgakopoulos, D., Shuster, H., Cichocki, A., Baker, D.: Managing Process and Fusion in Virtual Enterprises, Information Systems Vol. 26, No. 6, pp. 429-456, 1999

[Georgakopoulos 98] Georgakopoulos, D. (1998) "Collaboration Management Infrastructure for Comprehensive Process and Service Management" International Symposium on Advanced Database Support for Workflow Management, Enschede, The Netherlands, May 19th, 1998

[Georgakopoulos 95] Georgakopoulos, D. Hornick, M. Sheth, A. (1995) "An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure" Distributed and Parallel Databases, 3(2):119-153, April 1995

[Gibon 99] Gibon, P., Clavier, J.-F., Loison, S., (1999), "Support for Electronic Data Interchange, in Infrastructures for Virtual Enterprises" Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, 27-28 October 1999, Boston, pp. 187-208.

[Grefen] Grefen, P., Aberer, K., Hoffner, Y., Ludwig, H. CrossFlow: Cross-Organizational Workflow Management in Dynamic Virtual Enterprises

[Hoffner 98] Hoffner, Y, Schade, A. (1998) "Co-operation, Contracts, Contractual Match-Making and Bindings" 2nd International Enterprise Distributed Object Computing Workshop (EDOC 98), 3-5 November, San-Diego, USA

[Horsch 99] Horsch, A., Balbach, T.: Telemedical Information Systems IEEE Transactions on Information Technology in Biomedicine, VOL. 3, NO. 3, September 1999

[IBM] [9] http://www.ibm.com/

[Kaathoven 99] 4. Kaathoven, R. V., Manfred, A., Martin, S., Ulrich, R.: Organizational Memory Supported Workflow Management, Organizational Memory supported Workflow Management. In Proc. 4th Intl. Conference Wirtschaftsinformatik (WI'99), Saarbrücken, Germany, March 3-5, 1999

[Kim 01] Kim K.H., A Framework and Taxonomy for Distributed Workflow Architectures, Telecommunications Review, December 2001

[Koksal 98] 5. Koksal, P., Arpinar, S., Dogac, A.: Workflow history management ACM SIGMOD Record March 1998 Volume 27 Issue 1

[Koksal 98] 6. Koksal, P., Arpinar, S., Dogac, A.: History Management in Workflow Systems, international Symposium on Computer and Information Sciences (ISCIS XII), Antalya, Turkey, October 1998

[Krause 96] Krause, S., Magedanz, T. (1996) "Mobile Service Agents enabling "Intelligence on Demand" in Telecommunications", IEEE Global Telecommunications Conference (Globecom 1996), pp.78-85, IEEE Catalog No.96CH35942, ISBN: 0-7803-3336-5, IEEE Press, 1996

[Krause 97] Krause, S., Magedanz, T. (1997) "MAGNA - A DPE-Based Platform for Mobile Agents in Electronic Service Markets", IEEE International Symposium on

Autonomous Decentralized Systems (ISADS), pp.93-102, IEEE Catalog No.97TB100111, ISBN: 0-8186-7785-6, IEEE Press, 1997

[Krys 98] [15] Krys J. Kochut, Amit P. Sheth, John A. Miller ORBWork: A CORBA-Based Fully Distributed, Scalable and Dynamic Workflow Enactment Service for METEOR, Technical Report UGA-CS-TR-98-006, 1998

[Lazcano 00] Lazcano, A., Alonso, G., Schuldt, H., Schuler, C.: The WISE approach to electronic commerce. Int. Journal of Computer Systems Science & Engineering, 15(5), September 2000.

[Lindert 99] Lindert, Frank, Deiters, Wolfgang, Modelling inter-organizational processes with process model fragments, Proceedings of the Workshop Informatik '99, 1999

[List 00] 7. List, B., Schiefer, J., Tjoa, A.: Customer Driven E-business Process Improvement with Process Warehouse, IFIP2000, Peking

[List 01] 8. List, B., Schiefer, J., Robert, M.: Measuring Knowledge with Workflow Management Systems, DEXA'01, IEEE Computer Society Press, pp. 467-471, Munich, Germany, September 2001

[Maes 94a] Maes, P (1994a) “Agents that reduce work and information overload.” Communications of ACM.Vol.37. No.7. 1994. pp. 31-40

[Maes 94b] Maes, P. (1994b) “Modeling Adaptive Autonomous Agents”, Artificial Life Journal, eds by C.Langton, Vol. 1, No 1&2, pp. 135-162, MIT Press

[Magedanz 99c] Magedanz, T., Glitho, R. (Eds.) (1999c), Special Issue on “Mobile Agent-based Network and Service Management“ in Journal of Network and Service Management (JNSM), Vol. 7, No. 3, Plenum Press, New York, 1999

[Maij 00] Maij, E., Reijswoud, V.E., Toussaint, P.J., Harms, E.H., Zwetsloot-Schonk, J.H.M.: A Process View of Medical Practice by modeling Communicative Acts, Methods of Information in Medicine. Vol. 39, No.1, pp 56-62, 2000.

[Maij 00] [1] Maij , V.E. van Reijswoud 2 , P.J. Toussaint 3, E.H. Harms 4 , J.H.M. Zwetsloot-Schonk 5, A Process View of Medical Practice by modeling Communicative Acts, Methods of Information in Medicine. Vol. 39, No.1, pp 56-62, 2000

[Malone 91] Malone T W and Rockart J F (1991) Computers, Networks, and the Corporation. Scientific American vol. 265, no. 3. pp. 128-136.

[Marshak 94] Marshak, R. T., “Workflow White Paper – An overview of Workflow Software, “workflow ’94, San Jose, 1994

[Marshak 97b] Marshak , R. T., “Workflow: Applying Automation to Group Processes,” in Groupware – Collaborative Strategies for Corporate LANs and Intranets, D. Coleman, Ed.: Prentice Hall PTR, 1997b, pp. chapter 6, pp. 143-181.

[Microsoft] [7] http://www.microsoft.com/

[Miller 96] [12] Miller, J.A., Sheth, A.P., Kochut, K.J., Wang, X.: CORBA_Based Run_Time Architectures for Workflow Management System. Journal of Database Management, Special Issue on Multidatabases, 1996

[Miller 97] [14] Miller, J.A., Sheth, A.P., Kochut, K.J., Palaniswami, D.: The Future of Web-based Workflows, Research Directions in Process Technology Workshop, Nancy, France, July, 1997.

[Mitrovic 99] Mitrovic, D., Hunter, I. Male, S. (1999), “Characteristics of Networked Enterprise in Global Construction” in Proceedings of ICE’99, International Conference on Concurrent Enterprising, The Hague, The Netherlands, 15-17 March 1999, Nottingham: University of Nottingham, ISBN 0 9519759 86 pp. 447-454.

[Muehlen 00] 9. Muehlen, M. Z., Rosemann, M.: Workflow-based Process Monitoring and Controlling Technical and Organizational Iusses, HICSS 2000

[Muehlen 01] 10. Muehlen, M. Z.: Process-driven Management Information Systems-Combining Data Warehouses and Workflow Technology, International Conference on electronic commerce research (ICECR-4), Dallas, Nov. 8-11, 2001, pp. 550-566

[Muth 99] 11. Muth, P., Weissenfels, J., Gillmann, M., Weikum, G: Workflow History Management in Virtual Enterprise using a Light-Weight Workflow Management Systems, Proc. of 9th International Workshop on Research Issues in Data Engineering (RIDE), Australia, March 1999

[OMG 98] OMG jFlow Submission, ftp://ftp.omg.org/pub/docs/bom/98-06-07.pdf

[OMG 98] OMG, Minimum CORBA (1998), www.omg.org/docs/orbos/98-08-04.pdf

[OMG 01] [8] http://www.omg.org/cgi-bin/doc?ad/2001-06-09

[Orfali 96] Orfali, R. Harkey, D. and Edwards J. (1996) "The Essential Distributed Object Survival Guide" John Wiley & Sons.

[Ouzounis 99] Ouzounis, V. Tschammer V. (1999a) "A Framework for Virtual Enterprise Support Services", 32nd International Conference on Systems and Sciences (HICSS32) Maoui Hawaii, 3-5 January 1999

[Ouzounis 98e] Ouzounis, V. Tschammer V. (1998e) ”Integration of Electronic Commerce Business Processes in Virtual Enterprises” European Multimedia, Microprocessor Systems and Electronic Commerce Conference and Exposition, (EMMSEC 98) Bordeaux, France, 28-30 September

[Sadiq 00] 12. Sadiq, S. W., Maria, E. Orlowska: On Capturing Workflow Exceptions in Workflow Process Models, In Proceedings of the 4th International Conference on Business Information Systems. Poznan, Poland. Springer-Verlag. April 12 -13 2000

[Sheth 98] Sheth A.(1998) "Changing focus on interoperability in information systems: from system, syntax, structure to semantics" In Interoperating Geographic Information Systems goodchild, M., Egenhofer M., Fegeas R., Kottman C. (eds.), Kluwer.

[Sheth 97] Sheth, A. (1997) "From Contemporary Workflow Process Automation to Adaptive and Dynamic Work Activity Co-ordination and Collaboration". Proceedings of the Workshop on Workflow Management in Scientific and Engineering Applications, Toulouse, France, September 1997.

[Silver 94] Silver, B., “Automating the Business Environment,” in New Tools for New Times: The Workflow Paradigm, T. E. White and L. Fischer, Eds.: Future Strategies Inc, 1994.

[Spinosa 98] Spinosa, L.M., Rabelo, R., Klen, A.P. (1998), "High-level Co-ordination of Business Processes in a Virtual Enterprise", in Innovation, Agility, and the Virtual Enterprise, Proceedings of the 10th International IFIP WG 5.2/5.3 Conference, PROLAMAT 98, Trento, Italy, 9-12 September 1998, pp.725-736.

[Stricker 00] Stricker, C. Riboni, S. Kradolfer, M. and Taylor. J. (2000) "Market-based Workflow Management for Supply Chains of Services." In Proc. 33rd Hawaii Int'l Conference on System Sciences (HICSS-33), Maui, Hawaii, January 2000.

[Sun 99] Sun Micro System, Enterprise Java Beans –EJB Specification (1999) http://java.sun.com/products/EJB

[SWAP] Simple Workflow Access (SWAP), http://www.ics.uci.edu/~ietfswap/index.html

[Tagg 01] 13. Tagg, R.: Workflow in Different Styles of Virtual Enterprise, Workshop on Information Technology for Virtual Enterprises, ITVE 2001, Queensland

[Thompson 99] Thompson, S.G. Odgers, B.R. Shepherdson, J.W. (1999) "Cross Organisational Workflow should be Co-ordinated by Software Agents" Workshop on Cross-Organisational Workflow Management and Co-ordination February 22nd 1999, San Francisco

[Van 99] Van der aalst, W.M.P., Process-oriented architecture for electronic commerce and inter-organizational workflow, Information Systems Vol. ??, No. ??, pp. ??-??, 1999

[UML] [10] http://www.uml.org/

[Walton 96] Walton, J. Whicker, L (1996) "Virtual Enterprises: Myth and Reality", J. Control, Oct. 96

[Wargitsch 97] 14. Wargitsch, C., Wewers, T., Theisinger, F.: Workbrain: Merging Organizational Memory and Workflow Management System, University of Erlangen-Nuremberg, FORWISS, 1997.

[WFMC] [3] Workflow Management Coalition, http://www.wfmc.org/

[WFMC1] WFMC Interface1, Process Definition Interchange, http://www.wfmc.org/standards/docs.htm

[WFMC4] WFMC Interface 4, Interoperability, http://www.wfmc.org/standards/docs.htm

[Wognum 99a]Wognum, N., Thoben, K.-D., Pawar, K. S. (1999a), Proceedings of ICE'99, International

Conference on Concurrent Enterprising, The Hague, The Netherlands, 15-17 March 1999, Nottingham: University of Nottingham, ISBN 0 9519759 86.

[Wooldridge 95a] Wooldridge, M and Jennings, N. R. (1995a) “Intelligent Agents: Theory and Practice” The Knowledge Engineering Review. Vol. 10. No. 2. 1995. pp.115-152.

[Wooldridge 95b] Wooldridge, M., Jennings, N (1995b) Intelligent Agents: Theory and Practice. The Knowledge Engineering Review 10(2). pp.115-152. 1995.

[Zarli 99] Zarli, A., Poyet, P. (1999), “A Framework for Distributed Information Management in the Virtual Enterprise” The VEGA Project, in Infrastructures for Virtual Enterrises. Networking Industrial Enterprises. IFIP TC5 WG5.3 / PRODNET Working Conference for Virtual Enterprises (PRO-VE'99), Porto, Portugal, 27-28 October 1999, pp. 293-306.

[Zhao 98] 15. Zhao, J.L.: Knowledge Management and Organizational Learning in Workflow Systems, Proceedings of the AIS Americas Conference on Information Systems, Baltimore, Maryland, August 14-16, 1998