Case management system
design of Infopath-Yasper & case study: generic medical process

Liu, Hui

Award date:
2006

TECHISCHE UNIVERSITEIT EINDHOVEN

Department of Mathematics and Computer Science

MASTER'S THESIS

Case management system

Design of Infopath-Yasper &
Case Study: Generic Medical process

By
Hui Liu

Supervisor:

Prof. Dr. K.M. van Hee
Ir. I.G.J. Raedts

Eindhoven, November 2005

Abstract

Information systems have been developed to support managing business processes that are becoming more and more complex. Each new wave of technology has led to a new generation of information systems based on the principle of separation of concerns.

Since 1995 workflow management has become very popular. Workflow Management Systems (WfMSs) are currently offered by many software suppliers. The number of applications under development is growing, and users express the need for methods and tools to understand what can be expected in terms of benefits from such applications.

In this paper, case management systems are defined as a type of WfMSs that define, create and manage the execution of workflows, as well as manage the internal and external information related to the workflows. Conceptually a case management system is composed of a process definition tool, an administration tool, a client application, an information registration application and a workflow engine.

Regas is a case management system developed by H&B information systems. The study on this system shows that the data flow of the processes in Regas is well defined, but the control flow is not well managed. An alternative solution to enhance the control flow is to integrate Yasper and a workflow engine (YasperWE) to the system.

For the reason that H&B information systems cannot participate in the implementation of this solution, a case management system is designed using a Microsoft office application called Infopath instead of the Regas applications. As the process definition tools, Infopath is used to define the information model of the processes, and Yasper is used to define the behavior model. The other components of the system are developed during this project: Project Admin Control as the administration tool, Client Task Control as the client application, and YasperWE as the workflow engine. XML and web service technology are used to develop this system.

In addition, the performances of the system are evaluated based on an implementation on a case study: Generic Medical Processes. The result of the case study also provides a possible solution to handle hospital processes efficiently and to produce Electronic patient Record automatically.

Keywords:

Workflow Management System, business process, Enterprise Information System, case management system, Regas, Infopath-Yasper, Generic medical process.

Table of content

1 Introduction

1.1 Motivation

Information systems have been developed to manage business processes that are becoming more and more complex with every passing day. Each new wave of technology has led to a new generation of information systems, based on the principle of separation of concerns (Figure 1-1). [AH02][GH98]

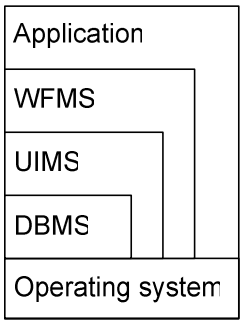

Figure 1-1 separation of concern

1965- 1975: *decomposed applications*. Information systems comprised decomposed applications, each with its own databases and definitions.
1975-1985: *database management system (DBMS).* The data management was first separated from the applications. Database is used as a shared memory that is defined by the DBMS. During that time, the data flow was defined in flowchart.
1985-1995: *user-interface management system (UIMS).* The user interface was separated from the applications. Since then the systems are developed to be more interactive and more user-centered.
1995-2005: *workflow management system (WFMS).* The business processes have been separated from the applications. WFMS became the new generation software components.

2005- : *Rule engine.* The Rule and business guidelines will be separated from the applications. Then the system will supports user's decision making based on formulized rules and guideline.

Business Process Re-engineer (BPR) was introduced in the early 1990s and caught the attention of management, because it promised radical changes and quick successes. It is based on the optimization of technology and work process alone. Reengineering became an extremely popular management change approach. [Davenport96] WFMS has been considered as the ideal tool for BPR, so it has been widely used in various fields, especially organizations that deal with tons of cases on daily bases, such as hospitals, governments, and social services organizations. WfMSs are currently offered by many software and hardware suppliers. The number of applications under development is growing, and users express the need for methods and tools to understand what can be expected in terms of benefits from such applications. One of them is H&B information systems B.V, a Dutch software company that developed a case management software package called Regas in 1996 that

was specifically built for welfare organization. A lot of improvements have been made since then, while their market has been growing as well. Their major customers used to be non-profit organizations that mainly operate in sectors such as: welfare, childcare, victim support and social care. Now they want to enlarge their market so they decided to cooperate with LaQuSo (Laboratory of Quality Software of Technische Universiteit Eindhoven) to find a scientific solution to improve Regas in order to handle more sophisticated processes.

1.2 Research objectives

The goals of this project are to define the characteristics of a case management system, and to suggest an alternative solution to enhance the performances of the Regas software package. In order to achieve the goals sub-objectives are defined:

1. *Define the main characteristics of a Case management system;*
2. *Find out what is missing in Regas*
3. *Find out a solution to improve Regas;*
4. *Develop a Case management system.*

1.3 Structure

Chapter 2 introduces some concepts of workflow management system. This includes a definition of a business process, an introduction of workflow management system, and a definition of a case management system and its structure. Besides some enterprise information systems are introduced and compared with workflow management systems. At last web service technology is introduce.

Chapter 3 will introduce Regas, including its architecture and strong- and weak- points.

Chapter 4 explains the alternative solution for Regas, and the motivation of the design of a case management system.

Chapter 5 describes the design of a Case management system that is based on a Petri net modeler, called Yasper and an XML editor, called Infopath.

In chapter 6 the Generic Medical Process case study is introduced, and implemented with the new case management system. The performances of the system are evaluated.

Finally, chapter 7 will address some conclusion and future work.

2 Workflow management system

The objective of this chapter is to characterize a good case management system. Therefore, in this chapter some definition of workflow management is explained. In addition the term *case management system* is used to define some software products, such as Regas, however this term has not been recognized as a scientific term. Therefore, it will be defined from the view of this project. Finally the characteristics of case management system are defined.

Sometimes the term *Workflow* is replaced by the term *Business Process*, nevertheless according to the definition given by Workflow Management Coalition (WfMC), they are not quite the same. WfMC defines workflow as "The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules."[WFMC99] So in order to understand the concept of workflow the following part the concept Business Process is discussed.

2.1 Business process

Definition

A Process [LM04] is a rather ambiguous concept with different meanings depending on the context in which it is used. In the management context, process is defined as "the transformation of input to output" [LM04]. In addition, Workflow Management Coalition (WfMC) defines a business process as "A set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships."[WFMC99]

Characteristics

Here we emphasis some characteristics of Business Processes: [HLUPIC03]:

- A business process realise implicit business objectives and goals;
- A business process involve internal and external knowledge;
- A business process realizes business objectives or policies;
- A business process consist of consist a set of automated activities
- A business process has defined conditions triggering its initiation in each new instance and defined outputs at its completion.
- A business process may involve formal or relatively informal interactions between participants; its duration may also vary widely.

A business process is a sequence of activates, which are structured according to certain business logic, and by executing these activities a certain business objective is achieved.

Generally these activities are deployed around one object also called *case*, and some staff members are assigned to execute them.

Take the *hospital billing process* shown in table 2-1 as an example: a process instance starts when a bill is "produced", then activities like "check insurance policy", "get billing address", "send bill", "send reminder", and "receive payment" are carried out by secretaries, billing managers and finance officers. Each piece of bill is an independent process instance, which should not be influenced by other instances. All those activities server the same objective: getting the payment.

Business Process	Business Objective	Case	Frequency of Occurrence	Period
Generic medical process	Cure patient	Patient	Daily	1 day to years
Billing process	Get paid	Bill	Daily	Days to weeks
Personnel training and evaluation process	Improve services quality	Personnel	Monthly	Weeks
Medicine purchasing process	Maintain daily operation	Drug order	Daily	Weeks
Equipment purchasing process	Maintain daily operation	Equipment order	Yearly	Weeks
Medical research process	Improve services quality	Research	Monthly or yearly	Years

Table 2-1 Example business processes of a hospital

Needs of automation

It is essential for an organization to use the minimal resources and time to accomplish the maximal business objective. Therefore it is very important how the resources are utilized to execute business processes effectively. However to manage business processes is difficult, because:

- some business processes starch for a very long period even years, like the medical research process example in table 2-1;
- some business processes occur frequently like the billing process, and normally a business process involves many people from different units in the organization;
- most of the time one single user may be involved in different executions of the same process, or even several different processes at the same time.

It is inefficient and barely possible to have some human managers to perform effective supervision the execution of the business processes. Therefore it is necessary to have an automated system to assist the execution of processes, so that every human resource knows when to do what task.

2.1.1 Map business processes to workflow models

To sum up the concepts defined so far: a workflow is the automation of a business process, and a workflow management system is the tool that supports the management of a workflow, and controls the execution of the business process.

In practice, there are many methods and tools that support modeling business processes to workflow. [AH02]

In the following some very important issues that influence the mapping from business processes to workflow models are discussed.

Control flow and Data flow

To define a workflow both control flow and data flow should be addressed:

The control flow is defined in a workflow via defining individual activities, their scope, the order of execution that maintains the overall business process logic, the rules governing the discipline of work list scheduling to performers, identification of time constraints and so on. The model describes the control flow of a process is called a behavior model.

Data flow indicates the movement of the information and the knowledge of a process. The movement should be addressed within every activity, and each case should contain a collection of knowledge. The model that describes the data flow is called an information model.

Workflow terms [PG03] [WFMC99]

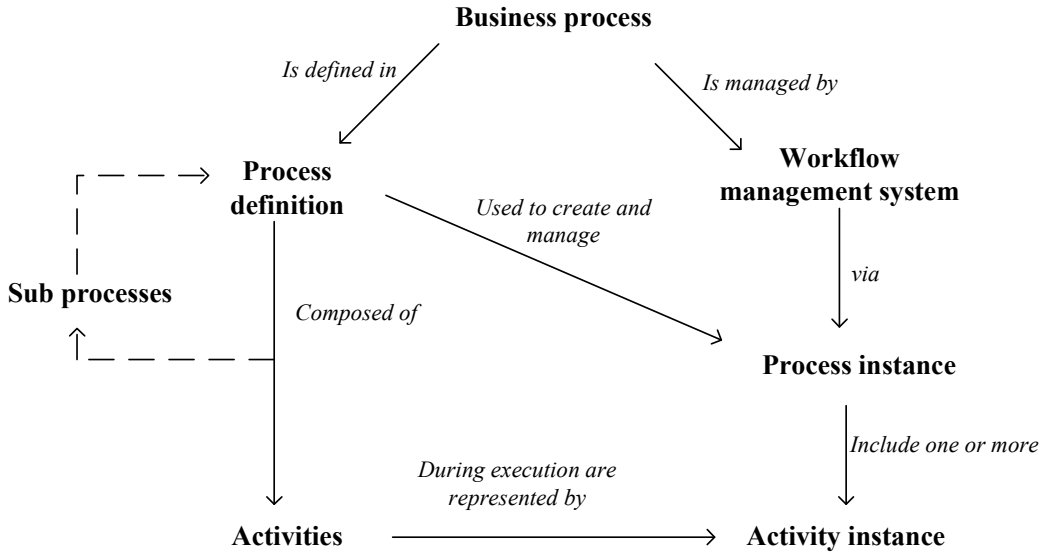

Figure 2-1 relationship between basic terminology [WFMC99]

Process definition:
A formalized view of a business process, represented as a co-ordinated (parallel and/or serial) set of process activities that are connected in order to achieve a common goal.

Case instance:
The representation of a single enactment of a process, or activity within a process, including its associated data. Each instance represents a separate thread of execution of the process or activity, which may be controlled independently and will have its own internal state and externally visible identity, which may be used as a handle, for example, to record or retrieve audit data relating to the individual enactment.

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

Activity instance:
The representation of an activity within a (single) enactment of a process, i.e. within a process instance. (See also general entry on Instance)

Participant:
One of the following types: a resource set, a specific resource, an organizational unit, a role (a function of a human within an organization), a human, or a system (an automatic agent). Answers the question "Who?" in a business process.

Workflow relevant data

Data that is used by a Workflow Management System to determine the state transitions of a workflow instance, for example within pre- and post-conditions, transition conditions or workflow participant assignment.

Knowledge object
Information that associates with certain processes, i.e. domain knowledge, guidelines, customer information.

2.2 Workflow management systems

2.2.1 Definition

WfMC defines a Workflow management system as:" *A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications*."
[WFMC99] It describes the basic function of a workflow management system:

- Import and export the process definition;
- Control the flow of the process according to the definition;
- Interact with the users;
- Invoke external applications;
- Communicate with other workflow management systems;
- Follow orders send by the administrator.

2.2.2 Workflow reference model

Figure 2-2 illustrates the workflow

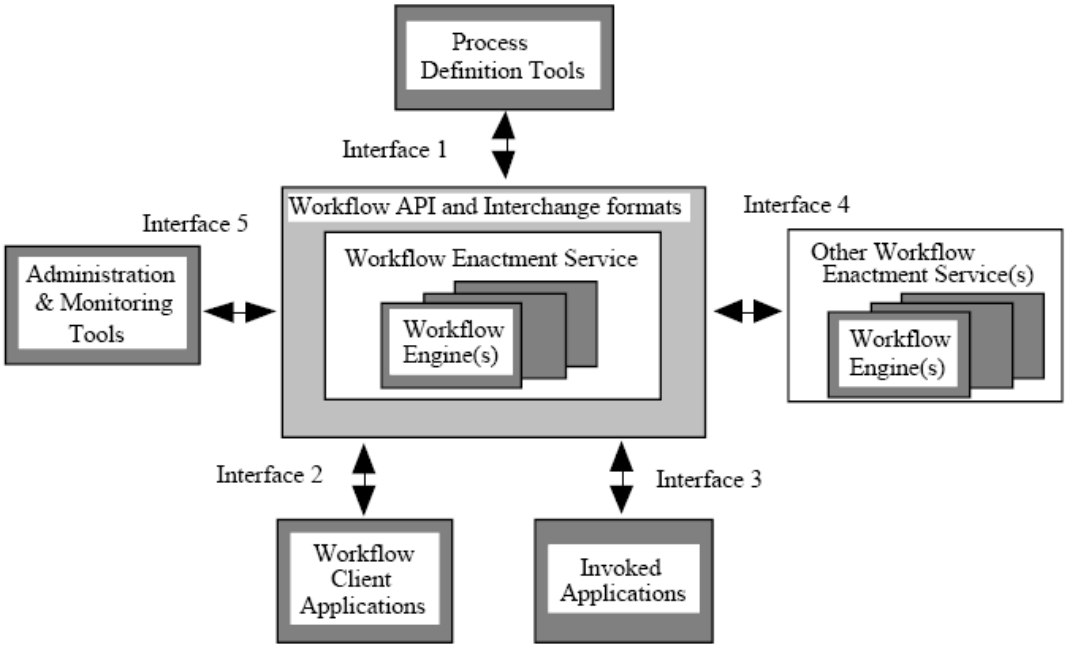

Figure 2-2 Workflow Reference Model

As shown in figure 2-1, the core of the product is the workflow engine, which has some main interfaces that communicate with other system components: the process definition tool and task handling tool, as well as the databases and external application. The descriptions of the major components are:

Workflow engine

A workflow engine is the core component of the workflow system, and it executes the interaction protocol according to the process definition. Sometimes a workflow engine cooperates with other internal or external engines to extend the services.

Process definition tool

Process definition tools are used to abstract the actual business processes in workflow models. They defines all the properties that are relevant to the workflow model.

Task hander

A task hander is a client application that distributes the tasks to the users according the workflow engine. It manages a work-list for each user, which indicates the work items that are needed to be done by this user.

Administration tool

Administration tool communicates with the system administrator and delivers the administrator's instructions to the workflow engine such as:

- Shut down the system;
- Launch the system;
- Create a new case;
- Remove an existing case;
- Assign a task to an employee;

External (invoked) application

An external application assists the users to execute tasks, which could be a data process application such as Excel, and Word, or any other information management system. Moreover, one workflow system can invoke more than one application, depending to the needs of the customer. Some workflow products build some application inside to provide richer functionality and better performance.

2.3 Case management system

The term Case management system is sometimes used for software that supports legal case management. However here we give a different definition: a case management system is a system that defines, creates and manages the execution of workflows, as well as manages the internal and external information related to the workflows.

A case management system can be seen as a combination of a workflow management system and a document management system, or a kind of workflow management system with information processing and document management function.

Case management systems are typically used to implement processes such as: Insurance Claim process; health care process, lawsuit services process; social services process and so on.

The common features of these processes are:

- Each case has a client. (the client can be a external customer or an employee of the organization)
- The procedure of a process depends on the information content of the case or runtime knowledge.
- The cases occur on daily bases and last for months or years.

- The information related to the cases needs to be reported and documented.

In order to handle the processes described above, a case management system should be

- Flexible: end user can adapt the processes according to the runtime decision
- Information oriented: supports data validation, data analysis and reporting

Different from normal workflow management system, case management system has internal information management tool that deals with the information model of the processes and relevant documents. A general Structure of a case management system is illustrated in figure 2-3.

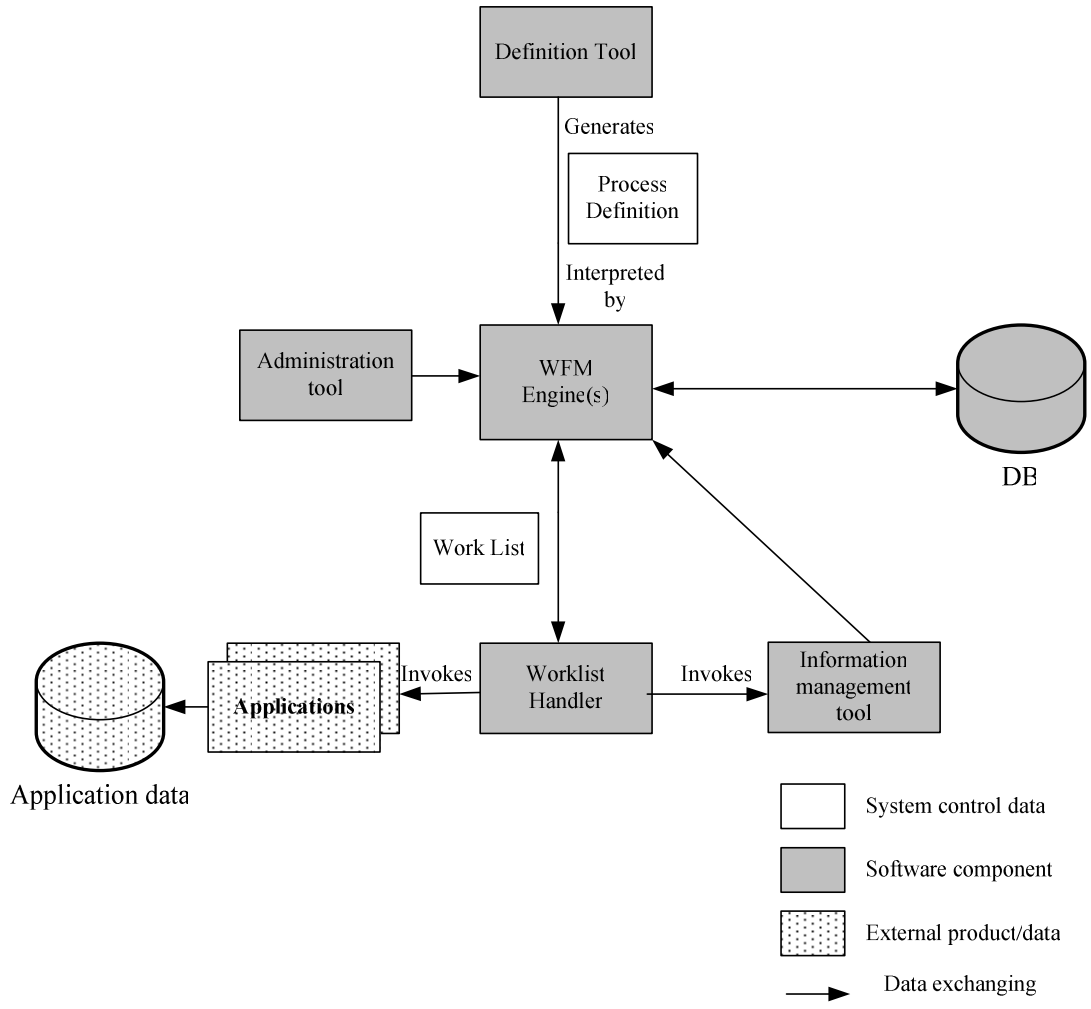

Figure 2-3 Structure of a case management system

2.4 Other Information Systems

Enterprise Resource Planning (ERP)

Enterprise Resource Planning (ERP) [WS04] [CP03] aims to obtain functional integration between the main business areas of an organization. ERP was intended to provide the connectivity and the common data models needed to link and coordinate the disparate functional areas within the organization, such as product planning, purchasing, logistics finance, etc.

ERP was originally envisioned as a "one-size-fits-all" modularized software approach to the management of most core business activities. Today companies use ERP to manage product planning, purchasing and logistics, inventory management, production, vendor management, customer service, finance, human resources and many other basic business activities. In addition, companies may also utilize best of breed applications for many of their specific needs. All of these systems must be integrated into and work with the core ERP system.

An ERP system consists of: a run-time environment, deployment tools, an application development environment and application programs. The latest technologies are used in ERP systems: Java, HTML, XML, EJB, CORBA/IDL, COM/DCOM, WSDL, JDBC/ODBC, legacy systems and so on.

Example product of ERP: SAP

SAP, which means Systems, Applications, Products in Data Processing, the company was founded in Germany in 1972. SAP is the market leader in ERP systems, and its main product R/3 is a standard ERP software system working with distributed Client-Server architecture. The modules of SAP include:

FI: Financial Accounting

CO: Controlling

AM: Asset Management

PS: Project Systems

HR: Human Resources

PM: Plant Maintenance

MM: Materials Management

QM: Quality Management

PP: Production Planning

SD: Sales and Distribution

CA: Cross Application

CA is latest developed module

- WF – workflow
- BW – business information warehouse
- Office – for email
- Workplace
- Industry solutions
- New Dimension products such as Customer Relationship Management (CRM), Product Lifecycle Management (PLM), Supplier Relationship Management (SRM), Advanced Planner and Optimizer (APO) etc

Customer Relationship Management (CRM)

Customer Relationship Management (CRM) is an integrated approach to identifying, acquiring, and retaining customers. By enabling organizations to manage and coordinate customer interactions across multiple channels, departments, lines of business, and geographies, CRM helps organizations maximize the value of every customer interaction and drive superior corporate performance [white05].

Nowadays, many companies try to establish and improve connections with their customers via CRM system, so to build long term mutually beneficial relationship. CRM is a comprehensive approach to improve the relationships with all kinds of customers, link front office functions (financial, operation, logistic and human resource) with customers (via such as internet, email, sales, direct mail, call center, advertising, fax etc.) When CRM is fully and successfully implemented, it is a cross-functional, customer-driven, technology-integrated business process management strategy that maximizes relationships and encompasses the entire organization [Goldenberg, 2000].

CRM has following benefits:

- It can extend the capability to the customer for self-service and Internet applications.
- It can attract existing and new customers through personalized communications and improved targeting.
- It can integrate customer and supplier relationships.
- It can construct metrics to analyze common and unique customer patterns.

As mentioned above, ERP tries to provide connections between all different areas of an organization. In fact, most of the nowadays successful ERP vendors pay attention on CRM market, such as Seibel, SAP etc., and try to build relation between CRM and ERP. The major difference between ERP and CRM is that ERP focuses on building foundation with tightly integrated back office functions, while CRM tries to link front and back office application to maintain relationships with customers to optimize customer

satisfaction and profitability. Although ERP is not required for a CRM system, it is beneficial for a CRM system if there is underlying infrastructure such as ERP.

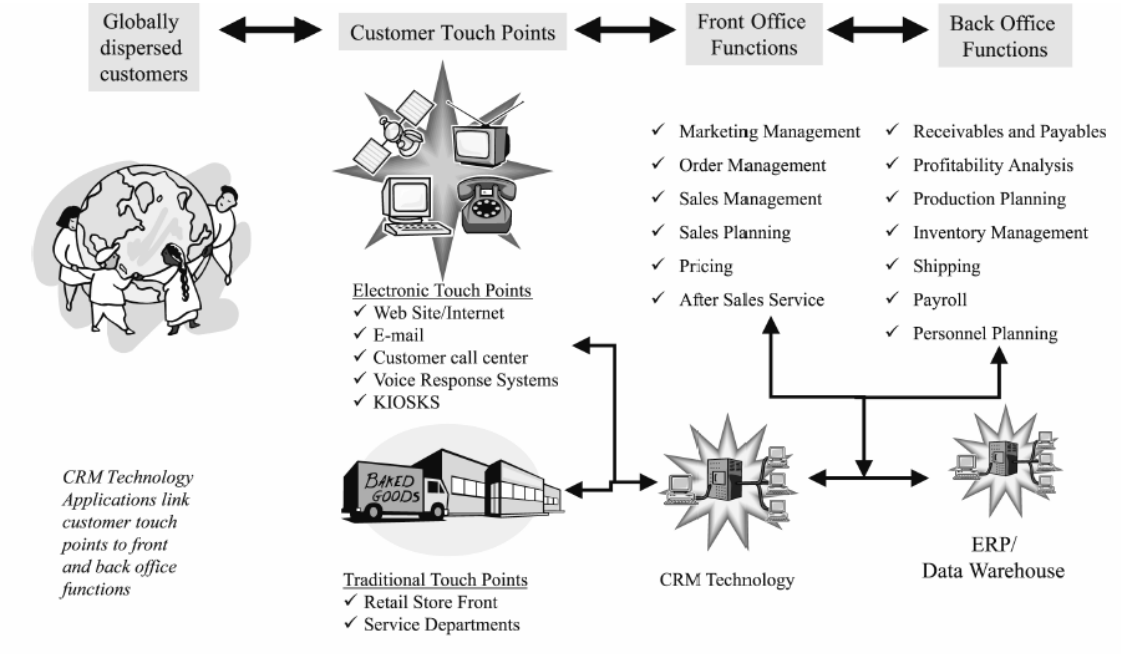

Figure 2-4 CRM applications supported by ERP/Data warehouse [CP03]

Although technology is a large portion of CRM, it is not everything. Managing a successful CRM implementation requires an integrated and balanced approach to people, process and technology. In addition to technology, CRM also should pay more attention to business process and people. While companies start to reengineer themselves around customers, their employees must also change their business processes, their organizational culture, and the way view customers and how to treat them.

Example product of CRM: Siebel

Siebel (which has been acquired by Oracle recently) is the leading CRM vendor that provides CRM and other solutions on helping organizations to build and optimize long-term relation around their customers. The solutions of Siebel on CRM include: [Siebel05]

Business Intelligence

Siebel provides business analytics as the business intelligence solution, which integrates data from multiple enterprise sources and transforms it into key insights that enable executives, managers, and front-line employees to take actions that lead to improvements in business performance.

CRM onDemand

The CRM onDemand try to provide easier solution to customer relationship strategy with little or no upfront IT investment. It provides businesses of all

shapes and sizes with a centralized repository of customer data and a real-time snapshot of all customer interactions on different CRM requirements.

Customer Data Integration

For the sake of customer data continues to be created, updated, and maintained in non-CRM systems, so customer information remains fragmented and duplicated across the organization, resulting in an incomplete view of the customer. Siebel customer data integration provides solution to centralize the management of customer data, provide complete picture of existing customer relationship, comply with stringent privacy, and improve risk management.

Sales Force Automation

Siebel also provides the solution on increasing sales by improving customer experiences at different stages of customer life cycle. The sales solution try to enabling the sales organization to share information across teams, manage sales pipeline, create quotes and proposals, configure products and services, also provide customer experience. It includes many functions such as sales analytics, mobile sales, sales forecasting etc.

Marketing Automation

It enables organizations to interact with their customers to understanding their unique needs and preferences.

Call Center & Service

The Siebel call center & service applications try to deliver customer service across customer touch points and cut service costs.

Partner Relationship Management

Siebel PRM applications provide companies with an enterprise-wide solution for managing relationships with their partner communities. It also increases channel revenue through improved collaboration, training, and communications.

Customer Order Management

Via product and catalog management, pricing management, product & pricing analytics, and quote & order life cycle management, Siebel customer order management applications try to reduce the complexity and cost of managing orders, so to accelerate the quote-to-cash process by creating relevant offers and promotions for customers.

Comparison of WfMS and EIS

	WfMSs	EISs (ERP &CRM)
Domain Scope	**Customized processes** WfMSs are used to define all kinds of business processes .	**Embedded processes** Workflow in EISs are hard-coded modules developed base on reference models or process templates
	Domain independence The same workflow infrastructure can be deployed to various domains.	**Domain specific** ERP and CRM systems are domain specific due to the adoption of reference models or process templates that embody the best business practices.
	Ad-hoc and dynamic domains *Ad hoc* and heterogeneous processes are better managed using WfMSs, mainly because they do not rely on predefined reference models.	**Static domains** ERP systems do not supply an effective framework for dynamic domains.
	No international settings WfMSs do not yet include some indispensable features, such as internationalization, multi-currencies, and multi-languages.	**International settings** ERP and CRM systems are well suited for international domains, since they offer features such as multi-language support, and multi-currency support.
Technological Scope	**Process-centric** WfMSs focus on the definition and execution of the processes	**Data-centric** EIS address only the transactional workflows, which are data oriented.
	Supports all kinds of workflows WfMSs handle workflows involving humans, IT applications, and transactional workflows	**Transactional processes** The underpinning of shared data structures across applications eliminates the need to pass data step-to-step among applications, which can now access data from a common structure. The focus is mainly on structured data transactions.
	Heterogeneous and autonomous environments WfMS are more suitable for heterogeneous, autonomous, and distributed systems	**Homogeneous environments with common data infrastructures** The ERP and CRM concept makes the strong assumption that data infrastructures are homogeneous across the organization, that is, that data is stored in interoperable databases, and in some cases, that databases used are all from the same vendor.
System Implementation	**Acquired as ready systems; Code automatically generated** Once workflows are designed, the deployment of applications is accomplished with little programming; the system automatically generates the necessary code for each application	**Based on pre-written "off-the-shelf" components** ERP and CRM systems are composed of prewritten software modules available "off-the-shelf"; they supply sufficient flexibility to match most organizations' needs by the setting of thousands of parameters. When a system module is acquired, it is fully deployed for a department.

	Bottom-up approach One of the starting points is to identify existing business processes – their logic, control flow, and data flow. The second step is for the organization to reengineer the process, if necessary. Once these steps are completed a workflow model can be constructed based on the information gathered.	**Top-down approach** One of the starting points is to acquire an ERP or CRM package to be deployed for a particular department. The second step is to set parameters to tailor the applications to specific characteristics and needs of the organization. The top-down approach forces organizations to follow external applications logic and policies.
	May require data conversions Workflow systems do not require a uniform and interoperable data infrastructure.	**Require data conversions** The platform defines architecture for data storage. The Legacy databases need to be replaced with ERP- or CRM-compatible databases.

Table 2-2 comparison of WfMS and EIS

2.5 Web service technology

Web Service represents the next generation distributed computing, building on and extending the current client-server model. Web service uses “loose coupling” concept, which means services are discoverable, platform independent and are expressed with self-describing interfaces, and can be invoked without external support. With these features, web service is able to provide flexible information sharing among people and business [W3C01] [W3C03] [PC05] [BF05]

For making the distributed computing easier and enhancing interoperability, web service uses eXtensible Markup Language (XML) as the means to express interfaces because it is platform neutral. Therefore, web service can be used as an XML-based messaging interface to computing resources that is accessible via internet protocols.

There are three main technologies developed for building web services: **SOAP** (Simple Object Access Protocol), **UDDI** (Universal Description Discovery & Integration) and **WSDL** (Web Service Description Language). [OASIS03]

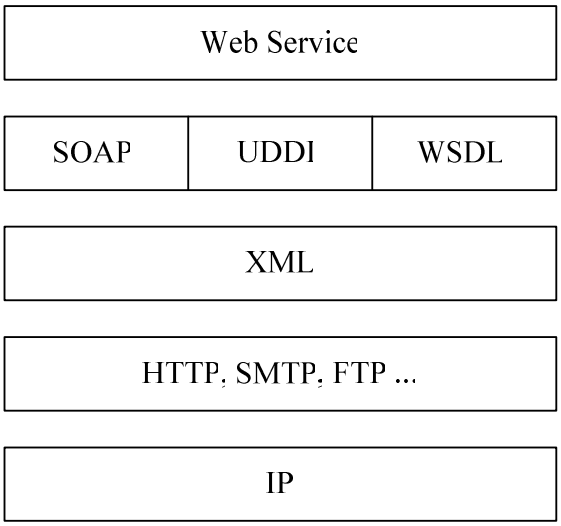

Figure 2-5 web service blocks

SOAP is a unidirectional, XML-based protocol for passing information. It supports the exchange of information (structured and typed) in a decentralized and distributed environment. SOAP is responsible for enabling platform independent interface to applications. Web service technology is not tied to a single transport protocol, although the common way to exchange web service request is via HTTP. However, other protocols such as SMTP or FTP also can support web service. SOAP message can be bound to underlying transport protocol such as HTTP or SMTP. The two main nodes in processing a SOAP message are initial message sender and ultimate message receiver. In addition,

SOAP intermediaries also play a role in SOAP message process for receiving message from sender and forwarding it to ultimate receiver.

WSDL is used to describe web service interface. A WSDL document describes a web service as a set of endpoint operation on messages containing either document oriented or procedure oriented information. It describes messages and operations abstractly. WSDL may describe what messages are input or output of certain operation, it also specifies the endpoints that provide the operations, and how to reach those endpoints such as via which network protocols.

UDDI provides a mechanism of how to discovery web service. For instance, a web service publishes its existence and interface via WSDL on UDDI server, where users can discover the web service via using UDDI registry.

3 Introduction of Regas

3.1.1 Overview

The Regas software package is developed by H&B information systems, and it is defined as a Casus management system by themselves. To be more precisely Regas is a mixture of a case management system and a CRM system for specific domain. It consists of two software tools: Regas_Admin and Regas_Client[1]. Regas_Admin is used to define processes, and Regas_Client is used to assistant the users to register information and analysis information.

The Regas software package has been used in the following sectors: Welfare, childcare, victim support and social care.

[1] To distinguish the software package Regas and the client application Regas, in this thesis the client application is renamed as Regas_Client

3.1.2 Terminology

The terms used in the Regas software products are defined differently from the terms that are used in WFM technical papers, table 3-1 lists the different terms used in the products and in WFM concept that are actually describing the same objects.

WFM	Regas_Admin	Regas_Client	Database
A group of processes	Process module	-	Processframe
Process	Case type	-	Workflowitemdefinition
Subprocess	Process item: Folder	-	Workflowitemdefinition
Task	Process item: Meeting/phone call/ letter/E-mail/Note/Analysis report/registration form/internal memo/registration form with obligatory client/adapter	-	Workflowitemdefinition
Content	Form, Question	Form, Question	Registration tab
Case type	-	Case type	workfidcaseid
Case	-	Case	Caseinfo

Table 3-1 Terms

3.1.3 System architecture

Use cases

Regas has two groups of users: the administrator and the employee.

The administrator can use Regas_Admin to define business processes by defining the workflow items and the relation among them. The administrator can assign resources to cases by defining function groups using Regas_Client.

The employee can register the case information and analysis the case information using Regas_Client.

Structure

Regas uses the client-server architecture, a database is built at the server-side, which stores the case definitions and information registrations. Regas_Admin and Regas_Client are installed on the client-side computers, and they both have independent interfaces and database management systems.

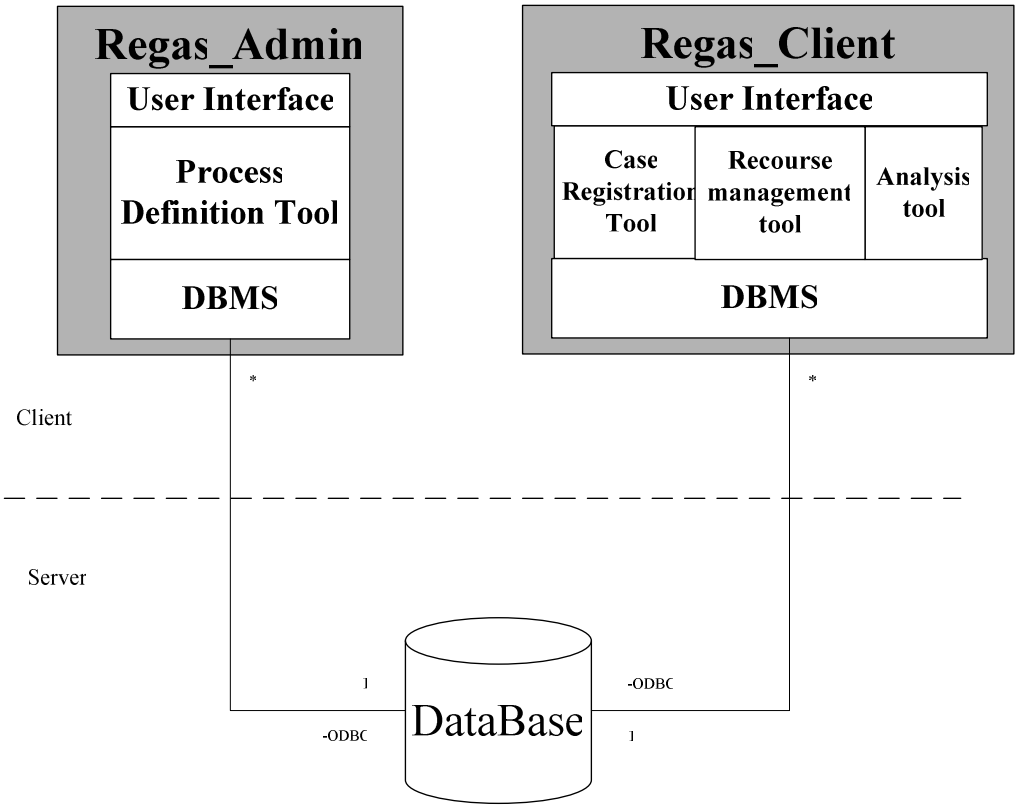

Figure 3-1 architecture diagram of Regas

Define a process

Information model:

Regas_Admin defines processes as process modules. Process modules consists case type, and a case type can contain folders or/and task items such as email, meeting, and notes. Each task item can contain several question forms, on which questions can be defined. Figure 3-2 shows the structure of the data.

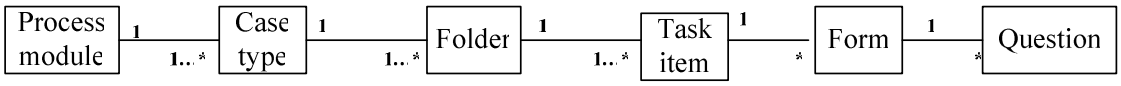

Figure 3-2 data structure of Regas

Control flow definition:

Regas_Admin uses workflow items' conditions to define the process. An activitie without any conditions can occur at any point of in execution time, and an activity with certain conditions can only occur when the conditions are fulfilled. The condition of each workflow item is defined from three aspects:

1. Relation with other activities
2. Relation with other folders or processes
3. Allowed occurrence times

Using these three kinds constrains Regas_Admin can define sequence and explicit choice patterns, but not be able to perform parallel execution, simple merge nor arbitrary cycle. Therefore, some processes cannot be defined properly on design time. Two counterexamples are shown in figure 3-3.

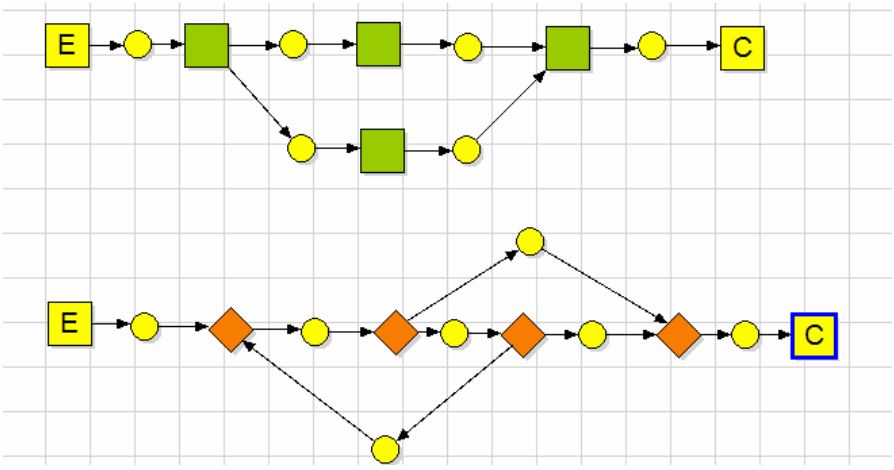

Figure 3-3 counterexamples

Nevertheless, the second example can be realized during runtime by the employee if the employee has the necessary knowledge of the process.

Function group (Role)

The takes are assigned to internal personnel by giving their function group the rights to access and edit these tasks.

Information registration

The employee can create new cases and register the information by answering pre-defined questions. Regas_Client stored the information in the database, and produces reports based on it.

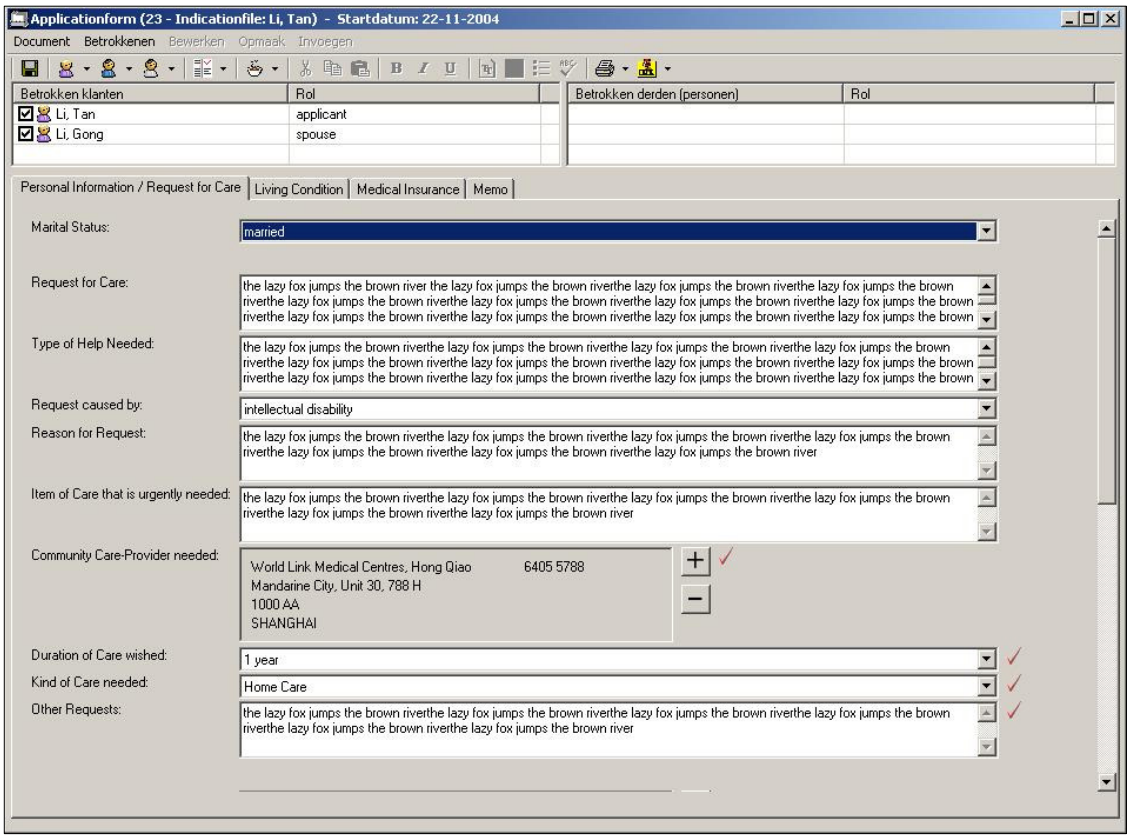

Figure 3-4 screenshot for information registration

Information analysis

The Stats-Factory that is included in Regas_Client, gives some possibilities to generate statistics on the data that have been registered. Regas can perform statistics based on frequency, activities, time spending and so on. Figure 3-5 shows an example analysis result.

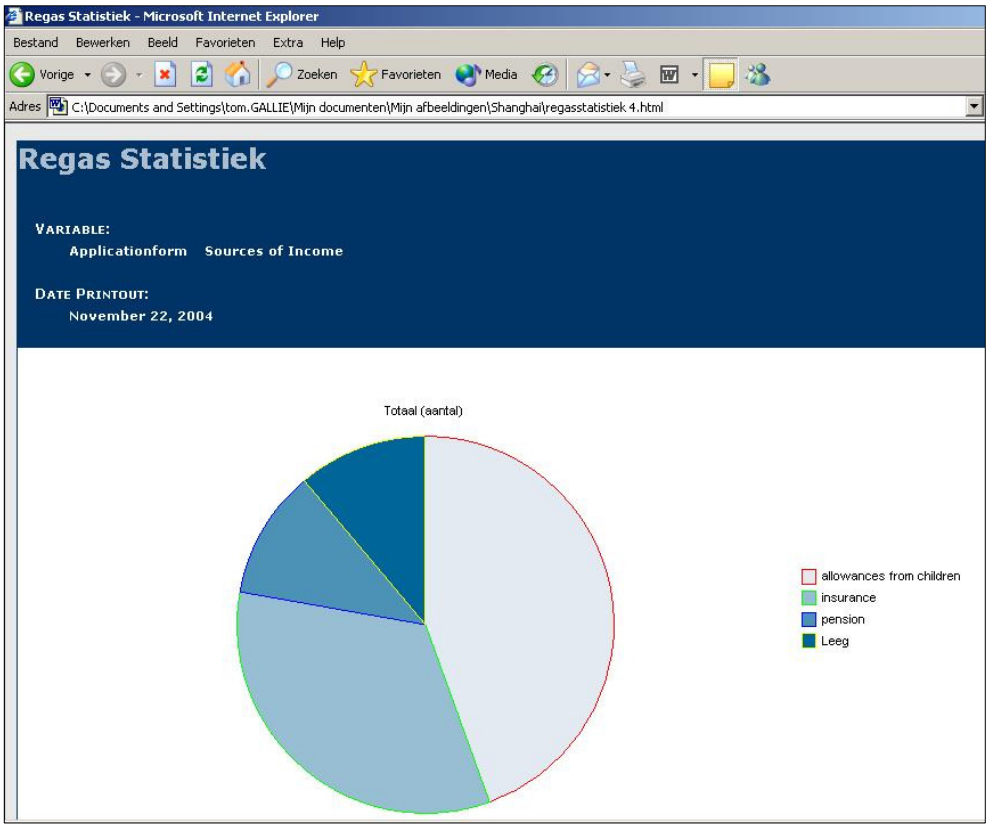

Figure 3-5 screenshot for the analysis result

3.2 Evaluation

Strong points

Regas contains some domain knowledge that is used by their customers, which makes it convenient for the customers to define a process and organize the information.

Various statistical analysis functions are provided based on simple querying the database via Regas_Client interface.

Regas supports process definition during runtime, which is sometime very useful, because in certain situations the procedure of a case is depends on the client's individual

states, and there is not a pre-defined module to follow, so the case owner (end user) should define parts of the process during execution.

Weak points

Regas leaves lots of control flow definition work to the end user (case owner), which increases the flexibility of process definition, however might lead to deadlocks or duplications and there is no validation during runtime to check and correct them.

The case information is stored in a database, and the information can be queried and modified. However, the execution of the cases is not really monitored or supervised by the system.

4 Alternative solution

4.1 Problem analysis

Regas provides stable information registration and statistic analysis, but the process definition function needs to be enhanced in order to handle more complex cases. Because the Regas_Admin cannot define enough constrains to thr process definition, errors such as deadlock or duplicates may occur during the execution.

In order to define the control flow of the process more precisely, an alternative structure is illustrated in figure 4-1.

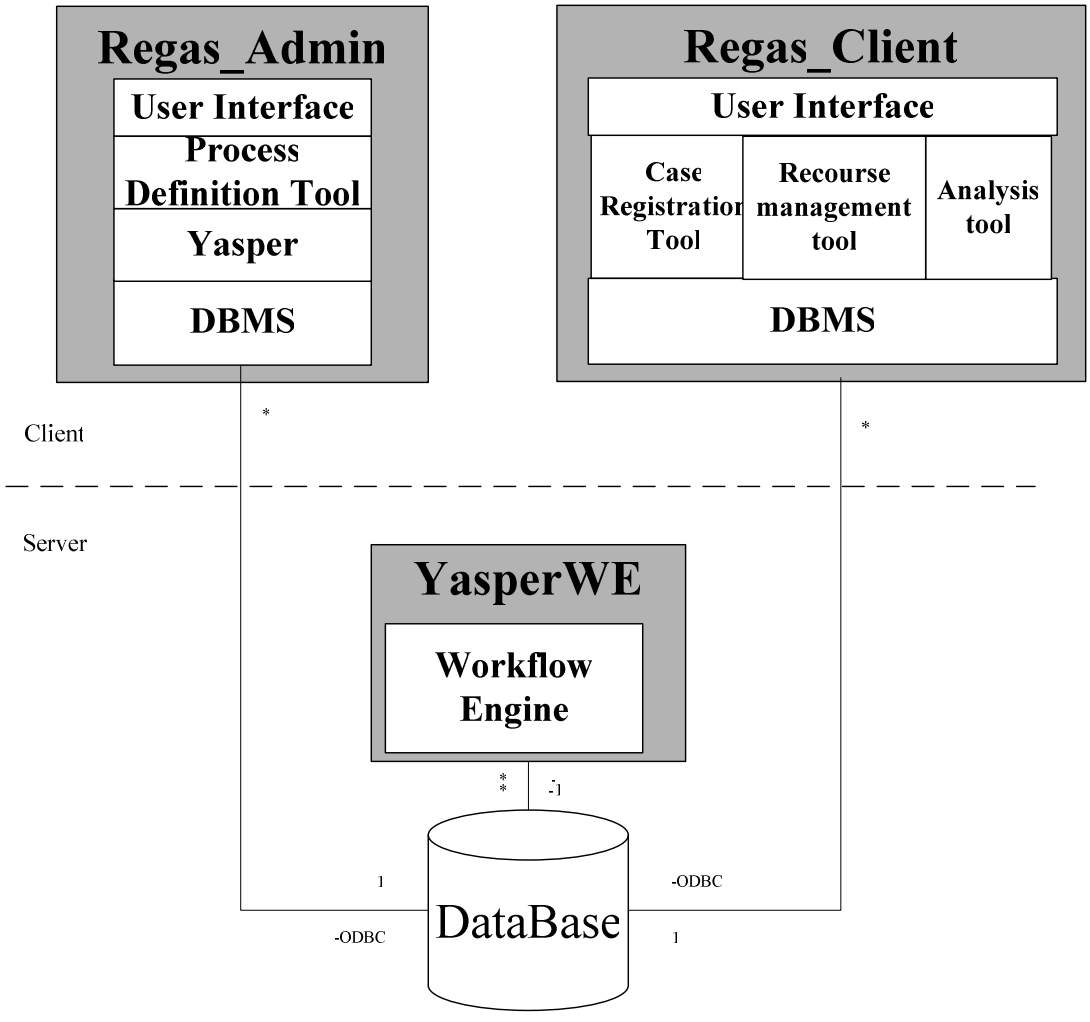

Figure 4-1 alternative structure

Yasper is a Petri net editor that should be used for modeling and simulating stepwise processes, and it is made for conceptual design and analysis. It maintains the control flow of a process that is modeled into a Petri net.

Based on the original applications, Yasper is integrated into Regas_Admin as a part of the process definition tool. The definition methods should be combined in such a way that Yasper models the control flow and the Regas_Admin definition tool defines the data flow.

Yasper will be extended to a workflow engine named YasperWE, which will be integrated with Regas applications to manage and handle tasks, and to monitor the consistency of the workflow.

4.2 Motivation of designing a case management system

H&B information systems cannot participate in the implementation of a solution at this point of time. Therefore, we decided to design a case management system using YasperWE. In this way, the Regas applications are replaced by other components, but Yasper and YasperWE work the same way as they should work in the alternative solution. So that the extension of Yasper can be realized, which is useful for the implementation of the Regas alternative solution. In addition, it is interesting to apply the concept of case management system to practice to see how well it can integrate the data flow and control flow of the real business processes, and whether the system brings values to the organizations.

4.3 Design a case management system

4.3.1 Extend Yasper to YasperWE

There are some changes need to be done in order to extend Yasper to YasperWE.

First of all, the firing action in Yasper should be decomposed into steps. When a transition is executed it moves the token(s) from input place(s) to the output places, and during the execution process a user needs a certain period of time to accomplish the task and fill out the form. In order to synchronize with InfoPath a transition is considered to have three sub-transitions: A, B, and C, as shown in figure 4-2

-Sub-transition A:

Consumes the token(s) from the input place(s), and modifies the status of the task from available to busy in order to avoid being picked again by other users.

-Sub-transition B:

Waits for the user to execute the task.

-Sub-transition C:

Receives the completed form and produces token(s) to the output place(s).

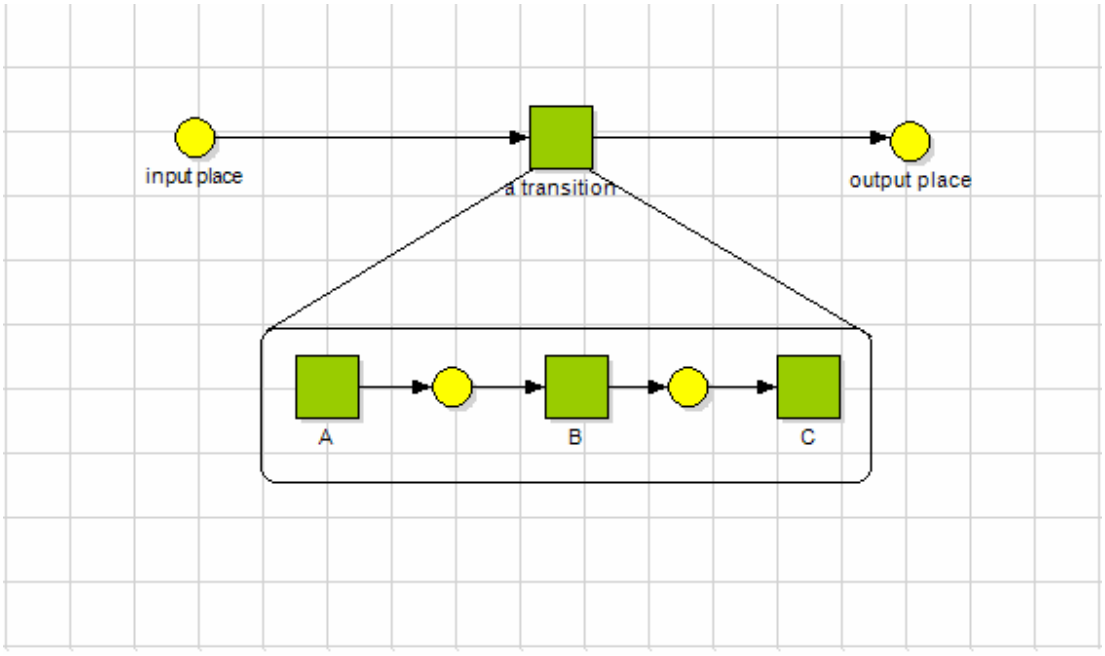

Figure 4-2 Subtasks of transition

Another important change needed to be done into make Yasper a workflow engine is that the cases that chandelled by YasperWE not only have case identities but also real content. In order to support the modeling, execution, and monitoring of those cases, YasperWE should not only be case sensitive but also content-aware. So a new content, case object, is introduced to YasperWE.

The characteristics of Case object are:

- Be associated with the each token in the Petri net;
- Contain the data content of each case;
- Be updated every time when a transition is executed;
- For one case type the content should be organized following the same structure;
- Be stored and organized properly so that it is easy to be tracked and monitored;
- Content should be easy to retrieve and analyze;
- Easy to integrate with other systems.

Due to those facts, XML is used to define the case object, and each case has its own unique XML file. This file is created as soon as the case starts in the process. When a transition is executed the system reads from this XML file, retrieves necessary data, and writes to it according to the content of the task. In addition when there is a choice in the process, the route would be decided based on the data contained in the XML file.

XML is flexible and can be exchanged among systems easily. However the XML code is trivial and difficult to read or edit. In addition the XML technology such as SOAP, XSLT, and scheme are not well known by end users. However, the users should not hesitate to use XML, therefore an XML editor is needed in order to create a XML-code-free environment where the users are only aware of the content of the information.

4.3.2 Introduction of Infopath

InfoPath 2003 is one of the new additions to Microsoft Office 2003 that converts user input into new or modified XML, which can be very easily integrated with any backend system that can understand and work with XML. [DUB03] [I3S04] [DRA03]

Infopath has two separate modes: form design mode and form entry mode.

In form design mode, the user can create a new electronic from scratch or modify an existing sample template, or build a form from various data source such as MS SQL server, Access, Web server even XML files or XML schema.

The user can decide what components to be put on the form including labels and all types of input filed, and where to put them. There are dozens of control types that users can choose from, of which the properties and validation can be set. In addition, the advanced user can add VBScript and Jscript statements for more complex uses, therefore the form could be very flexible and customized.

In form entry mode, Infopath focus on its easy-to-use function, so that the normal desktop user only fill out electronic form without acknowledge the format of XML or the process of transformation. Auto-complete, data validation checking and error alerts can ensure the correctness and completeness of the data. After the field is filled, the form will be saved as an XML file.

Due to the fats that are mentioned, Infopath is chosen to be a component of case management system. In addition, since Yasper and Infopath are the major parts of the system, the system is named Infopath-Yasper. The design of Infopath-Yasper is explained in Chapter 5.

5 Design of Infopath-Yasper

5.1 Use cases

There are three major types of human actors that interact with the case management system: *the process designer, the project manager, and the end user*.

The process designer uses the system to:

- Define the process
- Create electronic forms for the users that actually execute the activities,
- Define the relation between the views of the form and the transitions.
- Define users' role, determining which task should be performed by which users.

The project manager creates new project types and starts and stops new running instance of the project types.

The end user picks an available task from a task list, then executes the actual task and in the meantime completes an electronic form in order to record the data related to the task.

Figure 5-1 shows the simplified behavior model of the system: the process starts with the designer define the Petri net model and the information model of the process, then the two models are coupled as the entire process definition; then the project manager can use the definition to start new project type, and then start or stop project instances; the end user can pick up task from a running project instance execute tasks and fill forms.

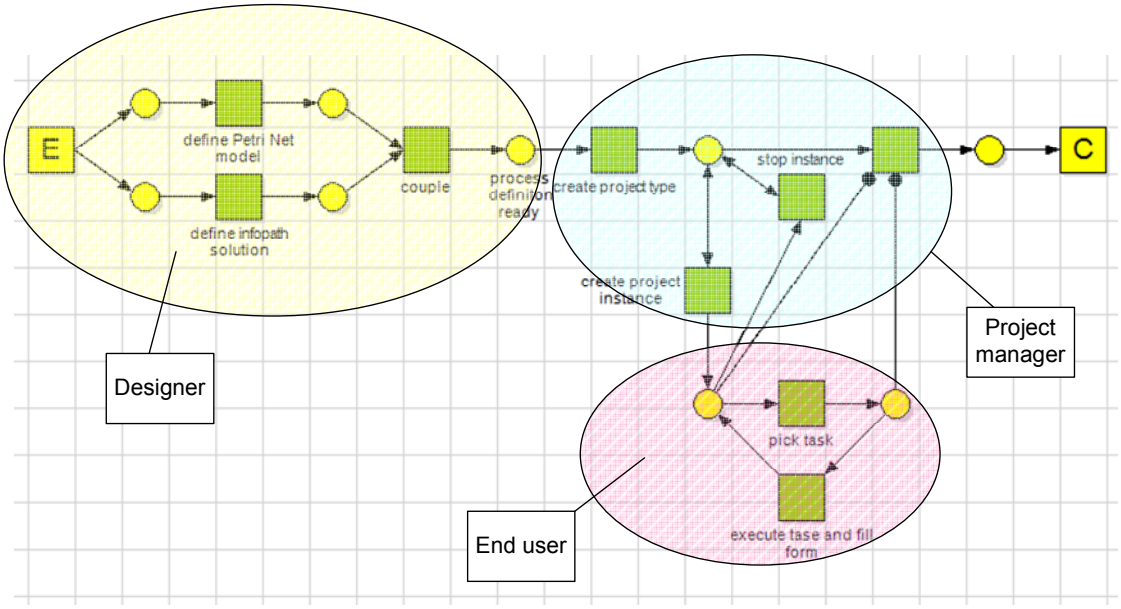

Figure 5-1 Petri net model of the system behavior

5.2 Conceptual structure

According to the discussion in chapter 2, a case management system should contain the following key components:

- Process definition tool
- Workflow engine
- Data administration application
- Administration tool
- Task-list handler

Figure 5-2 illustrates the conceptual structure of the case management system, and the components and the communication among them are described below.

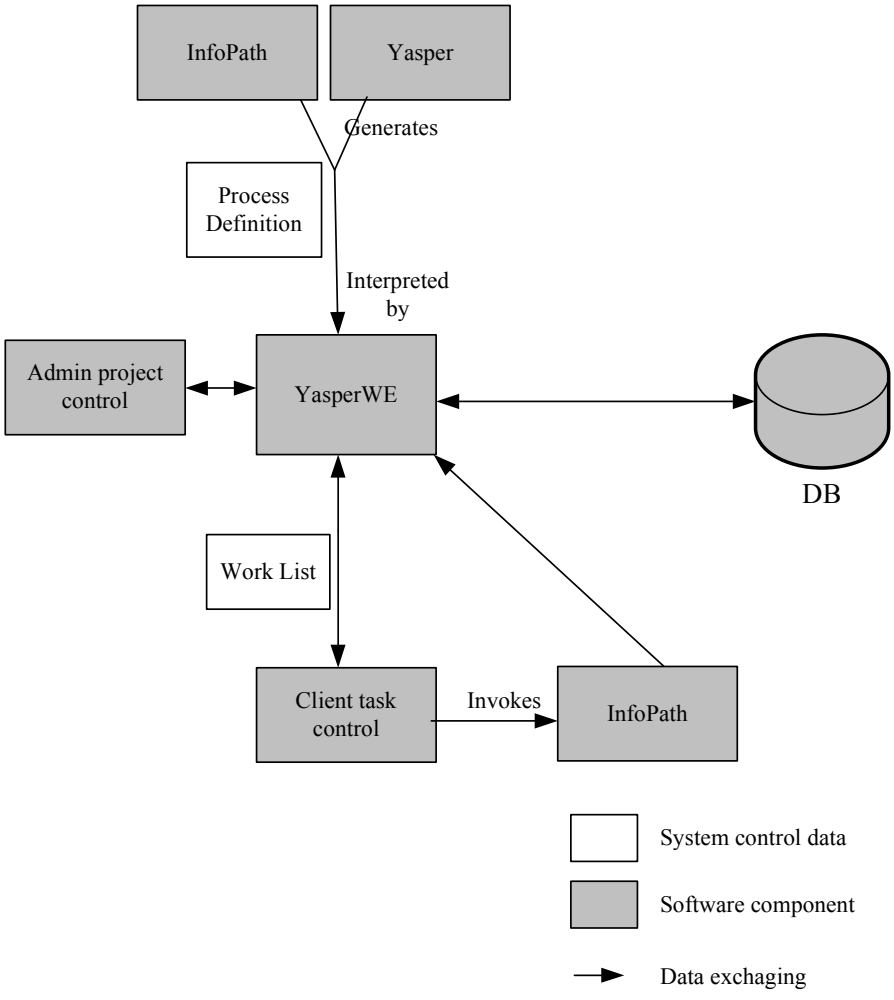

Figure 5-2 Conceptual structure diagram of Infopath-Yasper system

5.2.1 Components

Infopath-Yasper is consisted of five components: Yasper, Infopath, YasperWE, Admin project control, and Client task control. The first two are complete software tools that have been already developed, and the others are developed during this project.

Yasper

Yasper is a Petri net editor that supports the creation of business processes based on Petri nets. Yasper is a well-developed software tool and plays the role of process definition tool in the Infopath-Yasper system. It is used to define the Petri nets, classify resources and simulate the processes. (figure 5-3)

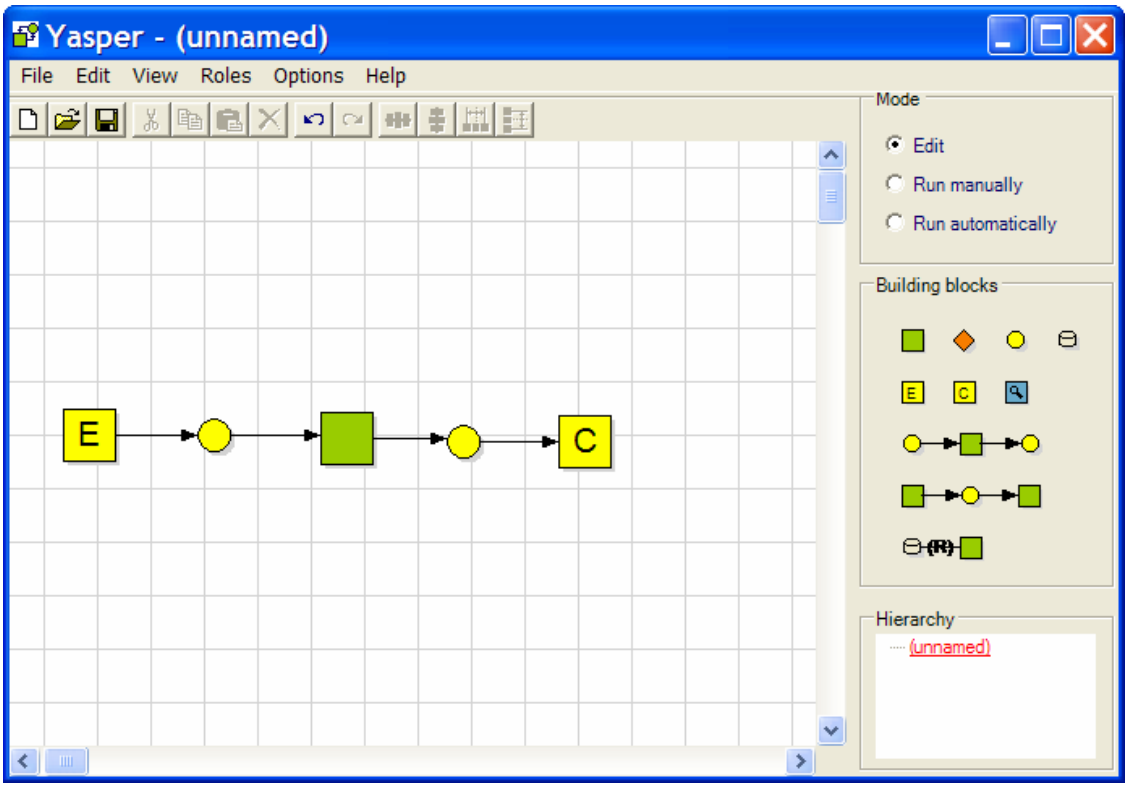

Figure 5-3 Interface of Yasper

The following functionalities of Yasper are used in the system:

Establish process definition

For each process definition a pnml file is created that contains the definition and the attributes of the process.

Define transitions

The designer can create transitions and define the properties transition. In the Infopath-Yasper system the description field has a special use which indicates which view from the Infopath solution should be used for this transition.

Define places

Places are allocated in-between the transitions, and indicates the states of the cases.

Define routing

The transition and places are structure into four basic constriction routings: sequential, parallel, selective and iterative, which represents the lifecycle of the cases.

Define roles

Roles can be assigned to the transition, which indicates the responsibilities of the resources. The designer can assign more than one role to do one transition and can define the capacity of the roles.

Simulation

Yasper provides two types of simulation function: manual simulation and automatic simulation, which can be used to check the correctness of the Petri nets as well as analysis the processes.

The Petri net model that defined with Yasper is combined with the solution file made by Infopath.

Infopath

InfoPath 2003 is one of the new additions to Microsoft Office 2003 converts user input into new or modified XML, which can be very easily integrated with any backend system that can understand and work with XML. [DUB03] [I3S04] [DRA03]

Infopath has two separate modes: the design mode and entry mode. In Infopath-Yasper the design mode is used as a definition tool, which defines the information model, and the he entry mode is used as the date administration tool.

Design mode

In form design mode, the user can create a new electronic. The user can decide what elements to be put on the form including labels and all types of input box, and where to put them. The user can apply rules and constrains to them in order to have more accurate input.

The outcome of such a design is called a solution file or a template file. It is a single CAB-compressed File with a file extension of .xsn. It includes include the .xsf file, XML Schemas, XML templates, and view, presentation, script, and custom business logic files.

The following table describes the form files and lists their extensions.[MSDN05]

Name	**Extension**	**Description**
Form definition	.xsf	An InfoPath-generated file that contains information about all of the other files and components used in a form. This file serves as the manifest for the form.
XML Schema	.xsd	The XML Schema files that are used to constrain and validate a form's underlying XML document files.
View	.xsl	The presentation style sheet files that are used to present, view, and transform the data contained in a form's underlying XML document files.
XML template	.xml	The .xml file that contains the default data that is displayed in a view when a new form is created.
Form template	.xsn	The compressed file format (.cab) that packages all the form files into one file.

Entry mode

In form entry mode, Infopath focus on its easy-to-use, so that the normal desktop user only fill out electronic form without acknowledge the format of XML or the process of transformation, more over using auto-complete, data validation checking and error alert can ensure the correctness and completeness of the data.

Admin project control

Admin Project Control is an administration tool that is developed for Infopath-Yasper. It supports the project managers' supervision and operational management of the workflow. The basic functions are:

Create project type

In order to create a project type the manager sends the name of the project, the Petri net model, the Infopath solution file, the initial XML file. Due to a limitation of Infopath, the Infopath solution file can only be stored on a certain location on the users' computers, so the project manager has to determine the location using admin project control.

Control project instance

The project manager can start or stop a new project type instance. (figure 5-4)

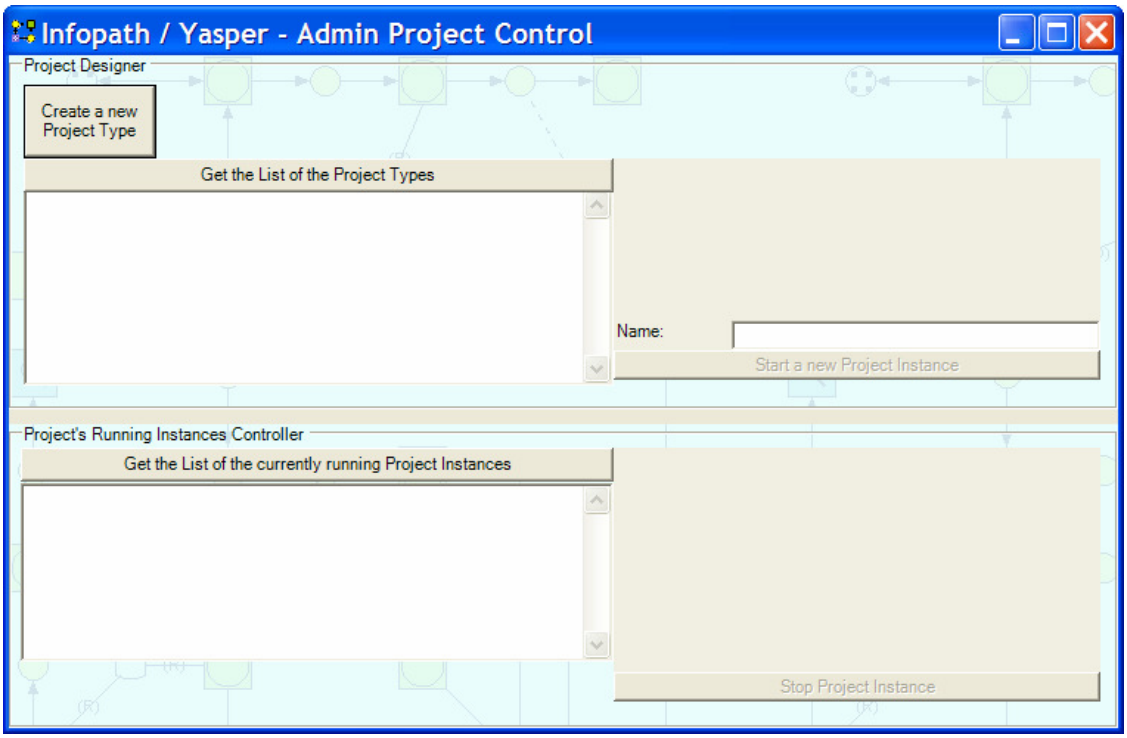

Figure 5-4 Interface of Admin project control

Client Task Control

Client task control is a standard task-list handler. When the end user logs in to the application, gets a task-list which shows which work items need to be carried out. By picking up a task, the user can start performing a task of a specific case. The users can manually send request to refresh the task-list or set the application to automatically poll for new task-list.

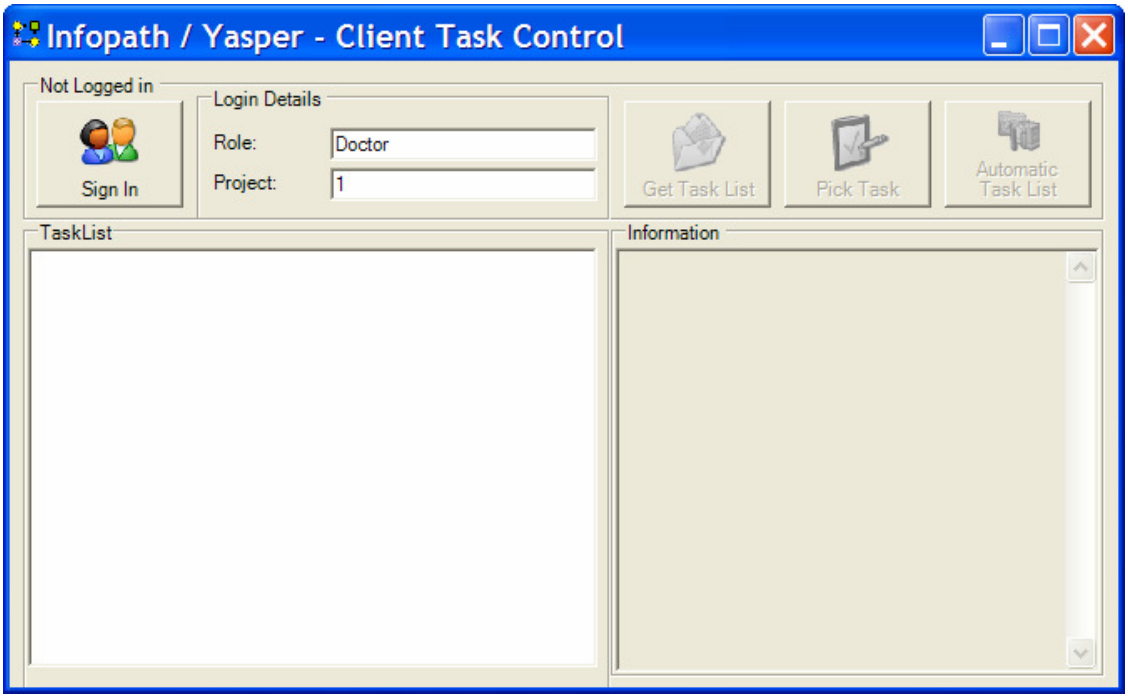

Figure 5-5 Client task control interface

YasperWE

YasperWE is the workflow engine of Infopath-Yasper. It uses some of Yasper's functionalities, a Database and a file system. It has interfaces that communicate with Admin project control, the Client task control and Infopath, but not directly communicates with any of the users.

5.2.2 Communication

Web Service architecture

Infopath-Yasper is consists of diverse client-side components: Infopath, Yasper, Admin project control, and client task control, and a server-side workflow engine, YasperWE.

All those components must work together to perform some tasks. Furthermore, the client applications operate in the different processing environment, so they must communicate by software protocol stacks over a network. So Web service architecture is used in the design. YasperWE is developed as a web service on the server side so that the client-side applications can communicate with the workflow engine.

The process definition tools: Yasper and Infopath are installed on the client-side. They are separated software that does not communicate with YasperWE, and only their process definitions are sent to YasperWE by Admin Project Control. Figure 5-6 shows the architecture of the system.

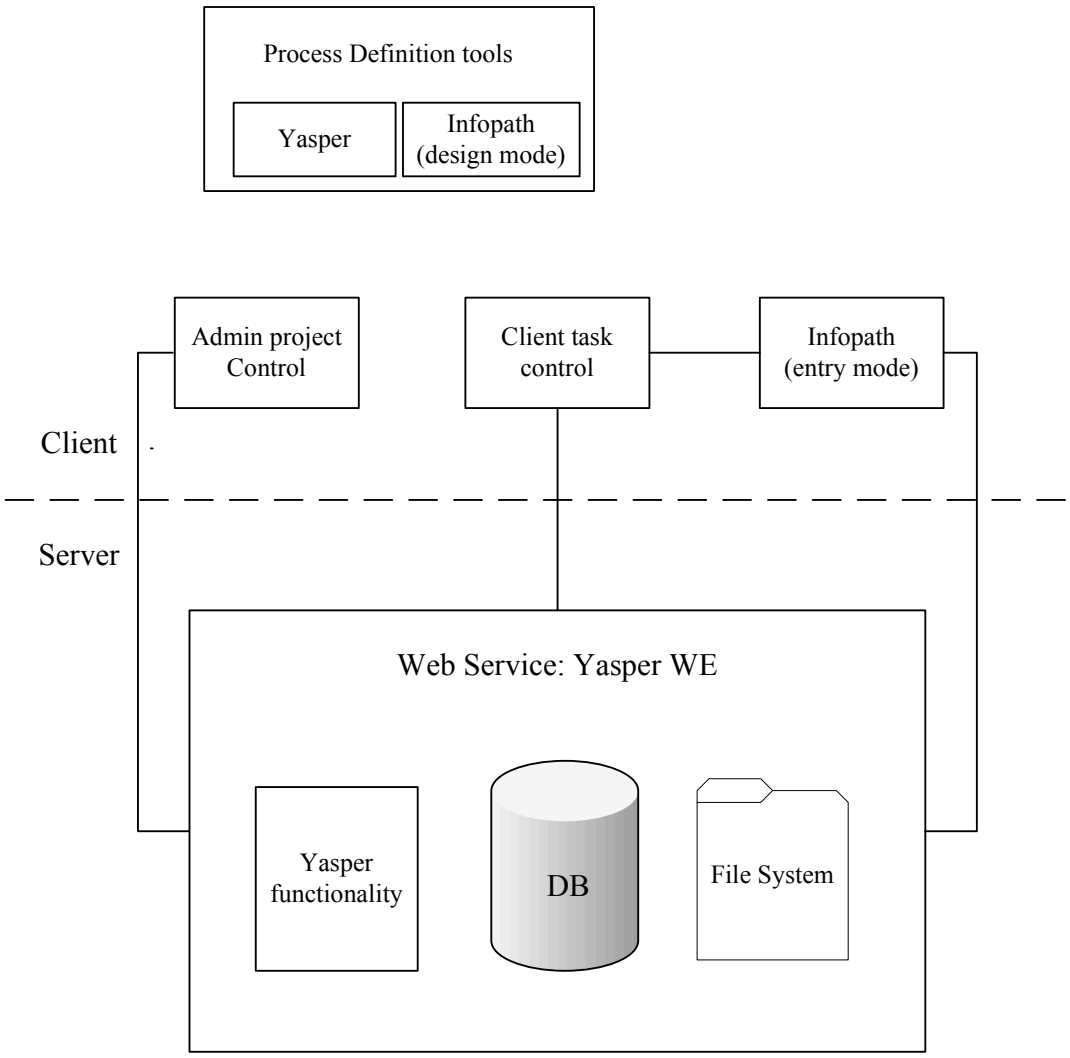

Figure 5-6 Infopath-Yasper architecture

Protocols

During the run time, the process of end user using the system is mapped to a Petri net model as figure 5-7. A user start the process with login the system then gets a task-list, so that the user can select a task from the list when it is not empty and executes that task. In the following the protocols of the three steps are describe.

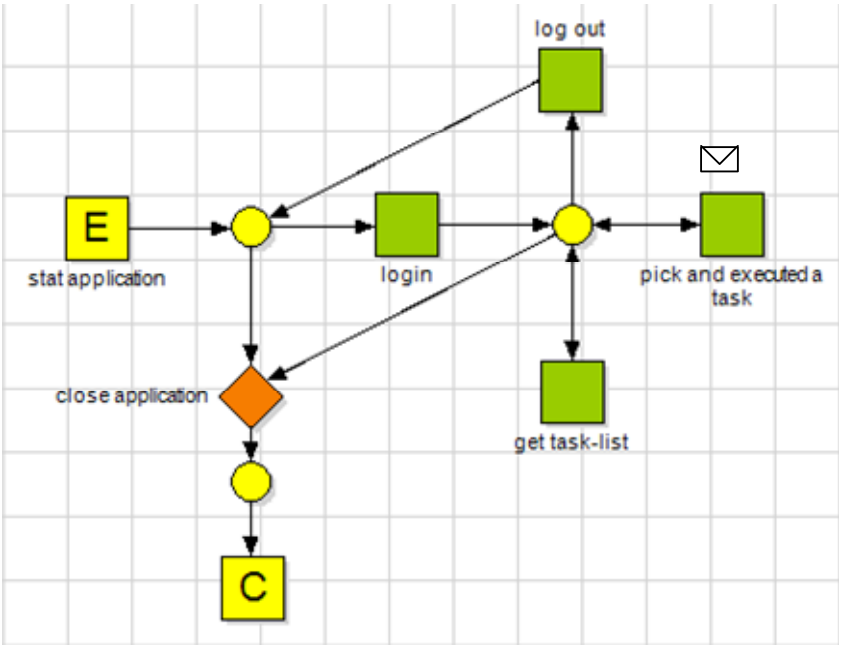

Figure 5-7 Process of execution tasks

Login protocol

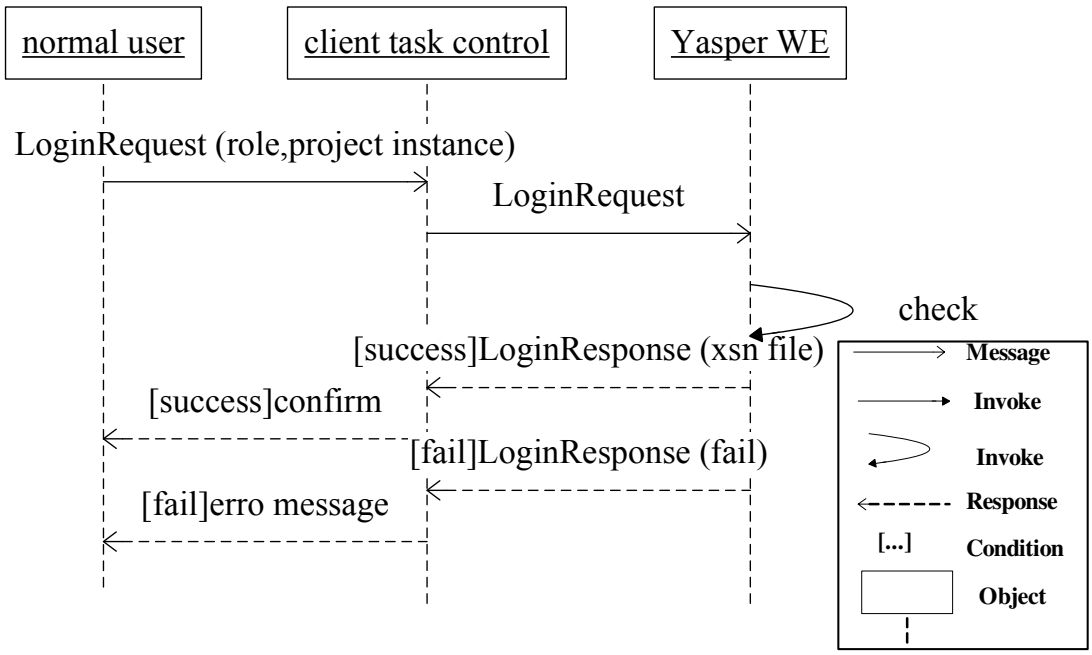

Figure 5-8 login protocol

The user fills out the role name and the project instance ID that the user wants to participates. Then Client Task Control sends a login request together with the role name and project instance ID to YasperWE. If the login information is correct, YasperWE confirms the request and sends corresponding Infopath solution file that corresponds to the project instance ID back to the client application. Otherwise, YasperWE sends a fail message back.

Get task-list protocol

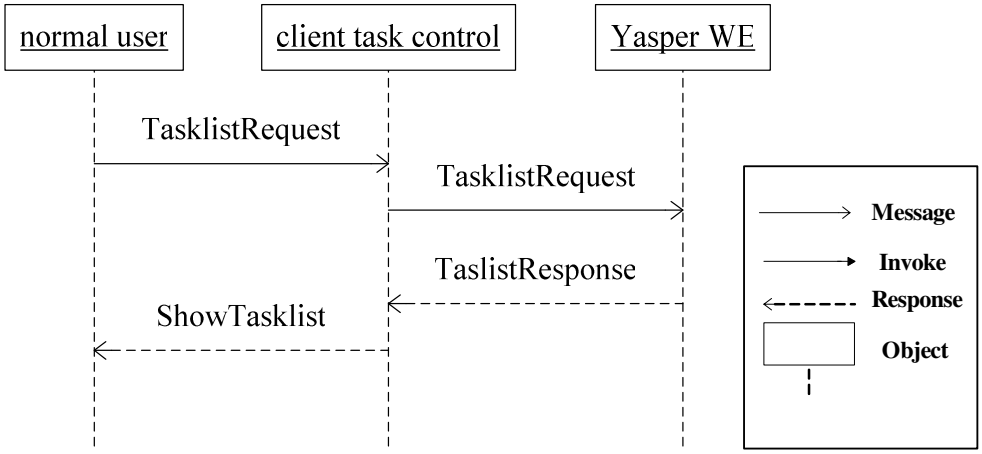

Figure 5-9 get task-list protocol

The user clicks the "get task-list" button on Client Task Control, the application sends a request to YasperWE uses the user role and the project instance ID. YasperWE generates a list of tasks that belong to the cases of this particular project instance and are available for this particular role, and sends it to Client Task control, which presents the list to the user.

Pick task protocol

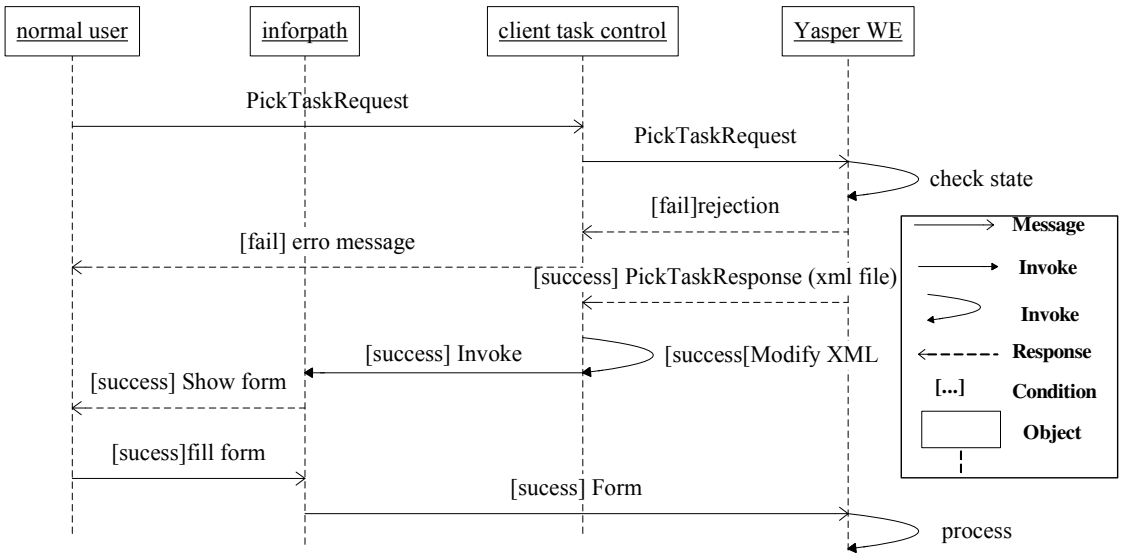

Figure 5-10 picking task protocol

The user picks a task from the task-list. Client Task control send a Pick Task request to YasperWE include the project instance ID and task ID. YasperWE checks the sates of the task, when it is no available, the request is declined and an error message is sent to the end user. If the task is available, YasperWE change the state to busy, so that other users can not pick this task again, then it finds the corresponding XML file of the case and the name of the view that is coupled with the transition according to the IDs, then sends them to Client Task Control. Client Task control modifies the XML file with the new task information and the name of the view that should be applied. It invokes Infopath entry mode, which resents the XML file as an electronic form using the indicated view. The user completes the form and submits it. Infopath sends the XML file back to YasperWE. Then YasperWE process task and moves the token to the outcome places. When the transition is an OR-split, the choice is made based on the user's input of the *Choice* filed in the form.

5.2.3 Critical decision:

Decision:

- When a user picks a task, the whole XML file is sent to the user's system.
- When a user submits an XML file, the file is used to replace the old one at the server.

Risks

Non-Parallel execution:

Figure 5-11-I is an example of a parallel execution. The emitor is an AND-split, once it is fired both task A and task B are available for execution. Task A and task B could be executed simultaneously, but there is a problem: assume that role1 picks up task A, while role2 picks up task B, and they execute the tasks at the same time and both fill out the form. Suppose that role1 submits the form first, the XML file of this case is rewritten including new information that is added by role1. When role2 submits the form, the XML file of this case is rewritten again, the new XML file contains the new information that is added by role2 but not the information added by role1. Therefore, a part of the information is lost during the Parallel execution. In order to avoid such a problem, the parallel execution is forbidden in the system. Figure5-11-II shows the solution for this problem, once the process starts either task A or task B can be executed first, but once one of them is executing by a user the other task has to wait until the task that is currently executing is done.

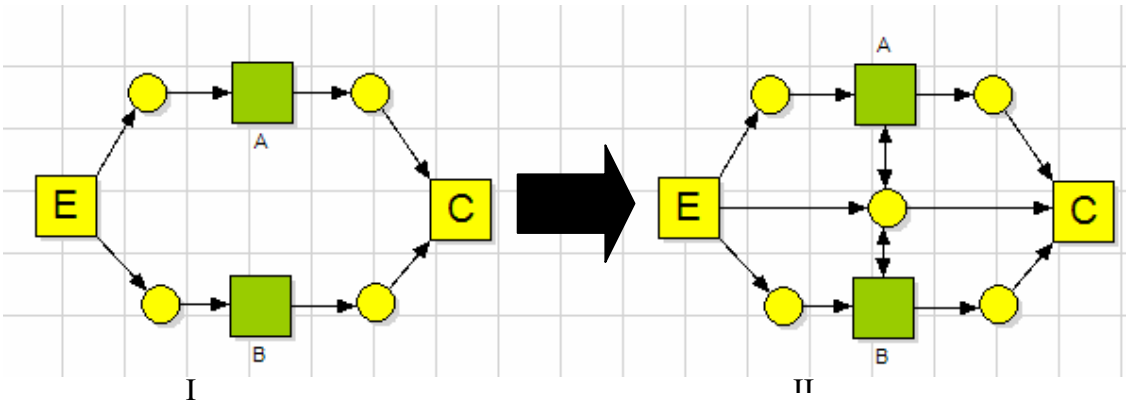

Figure 5-11 Transformation from parallel process to non-parallel process

Security issue: confidentiality violation

Sending the entire XML file to each user will expose the sensitive information to unauthorized access. In some cases the user with a certain role is only supposed to access to a part of the XML file, even though Infopath filters the data according to the view, the user can still access to the entire XML file via the file system.

Alternative Solution

An alternative solution is to only send parts of the XML file to the client side according to the content of each task, and update the original XML file at server side by merging the submitted and original XML file.

Merging XML file can be done by either InfoPath or another XML editor, nevertheless, the rules of merging should be defined specifically for each transition, since there are five actions that can be taken to perform a custom merge:

insert This action inserts data based on the specified XML node.

delete This action removes data based on the specified XML node.

replace This action replaces the data based on the specified XML node.

ignore This action ignores the specified XML node.

mergeAttributes This action combines the attributes based on the specified XML nodes.

The merging rules should define under which conditions which action should be applied at which elements.

Trade off

In order to realize custom merging, a lot of additional specification is required at design time, which results in increases of

- design time
- system complexity
- extra testing costs.

6 Case study: Generic Medical Process

6.1 Introduction

The purpose of this case study is to use Infopath-Yasper to handle a real business process with real requirements that are complex enough to use all the features of the system, and to find out what work should be done next to improve Infopath-Yasper .

A Generic medical process is the process that a patient has to go through at a nominal hospital. The reason of this process being chosen is that the procedure is well known, so it is easy to start with. Secondly, the result of this case study may contribute to the researches on improving medical process performances.

The problems that need to be solved in current medical system are:

- Low efficiency is the most noticeable problem. Sometimes the waiting time is unbearable for patients.
- Collaboration among organizations is difficult due to differences in guideline and use of applications.
- Disease management needs to be enhanced based on the use of Electronic Patient record.

The goal of Generic medical process is to provide high-quality and cost-efficient patient care, in other words to cure more patients with less expense in the same time period. In order to do that there are several things need to be considered:

- Guidelines should be represented in a structured format
- Processes should be modeled to reflect the guideline and decision rules
- Patient data should be mapped to Electronic Patient Record
- Resources should be organized to complete tasks within the shortest throughout time

In the following, an example Generic medical process is introduced and implemented with Infopath-Yasper. Then the performance is analyzed and evaluated.

6.2 Case description

The Generic Medical Process is the process that a patient has to go through at a normal hospital. Figure 6-1 shows the mapping of the workflow model to the knowledge environment of a hospital. Each patient's visit is considered as a case, which starts with the patient's arrival and ends when the patient is released. Each patient has own patient record. The process consists of a series of tasks such as diagnosis, tests and treatments. The tasks are performed by different groups of resources: doctors, test experts, therapists,

receptionists and schedule planners. The goal of the implementation is to map the organizational knowledge, clinical information and medical knowledge to a workflow definition, so that the activity can be managed more efficiently, and information involved can be registered for future reference. [PBTWOZ04]

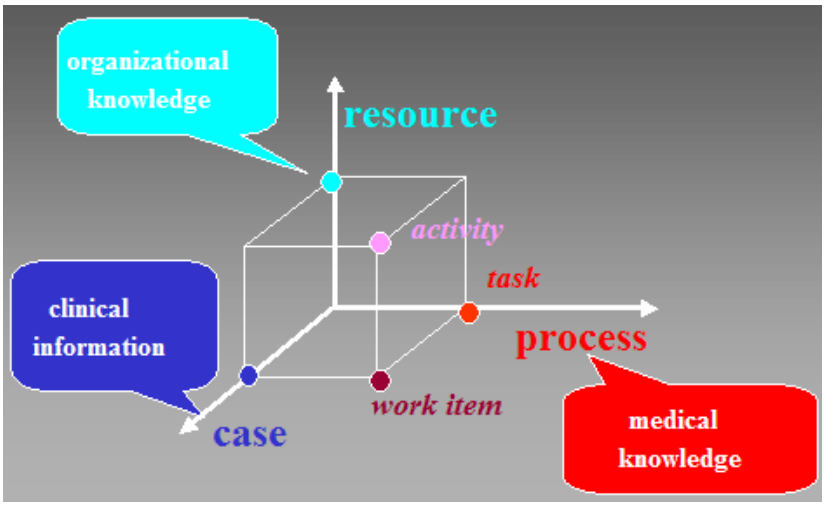

Figure 6-1 The mapping of the workflow model to the knowledge environment of a hospital

Process description

Generic medical process

The process (figure 6-2) starts with a receptionist registering the patient at the front desk after that the schedule planner makes an appointment for the patient for the initial diagnosis. The diagnosis (figure 6-3) is carried out by a doctor who will decide what to do with this patient. If the patient does not need any help or cannot be helped, the patient will be released, causes the process to end. If the doctor thinks someone else can help the patient better, the patient is referred to a doctor from another expertise. Another possibility is that the patient is asked to do a few tests (figure 6-6) before the doctor can draw any further conclusions.

There are many different types of tests, like biochemical tests, visual tests, and function tests. The tests are carried out by different test experts, and the test results are sent to the doctor for the follow-up diagnosis (figure 6-4). The doctor may release the patient, consult other experts, refer to other doctor, ask for more tests, prescribe certain therapy (figure 6-7), or therapy for the patient.

There are many different types of therapy, like medication, surgery, radiotherapy, and physiotherapy, of which therapists are in charge. After the therapy, there is another diagnosis (figure 6-5), after which the doctor can make decision based on the results of the therapies.

Besides, for any diagnosis, test, or therapy an appointment has to be made beforehand by the schedule planner. Figure 6-2 illustrates the how the process described above is define in a Petri net.

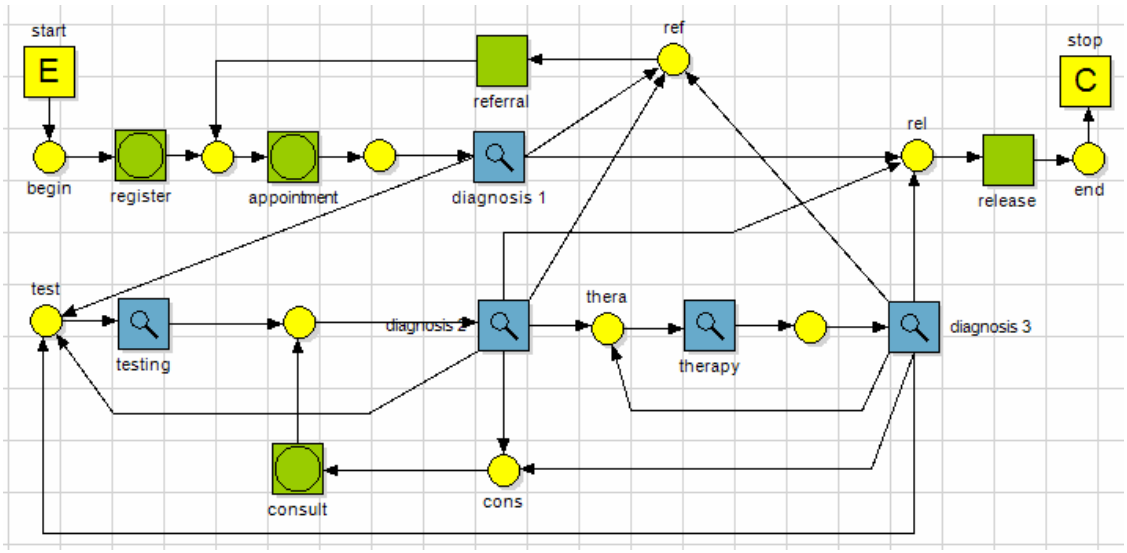

Figure 6-2 Generic Medical Process

The subnets are described as following:

Diagnosis 1

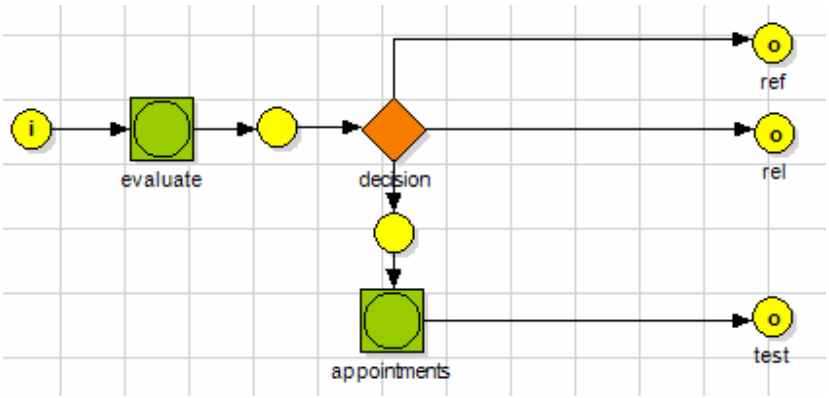

Figure 6-3 diagnosis 1

Diagnosis 2

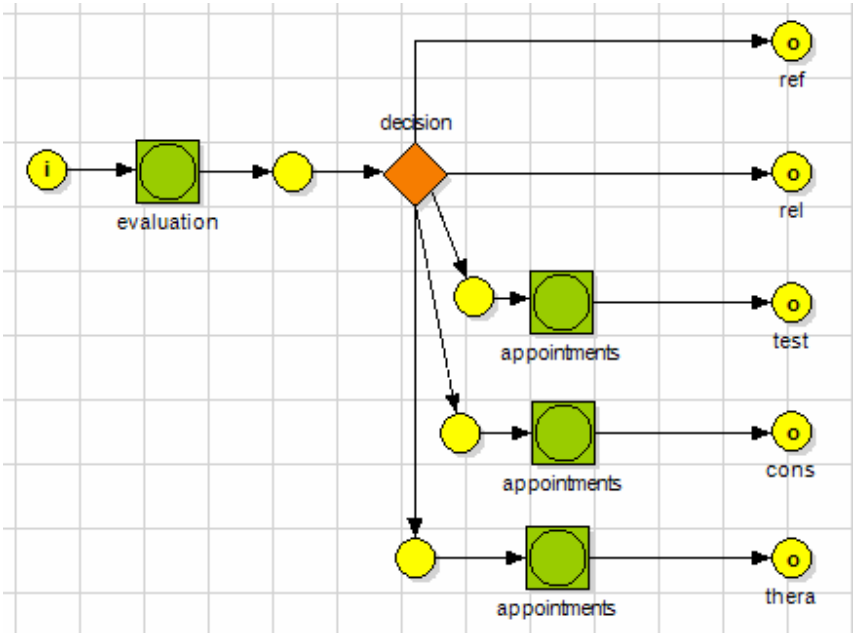

Figure 6-4 diagnosis 2

Diagnosis 3

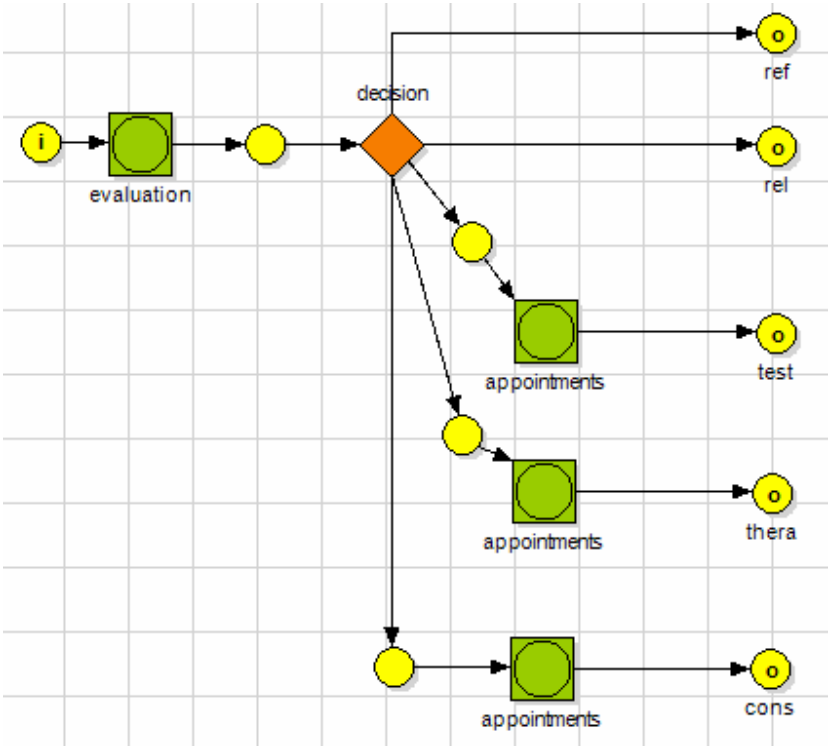

Figure 6-5 diagnosis 3

Testing:

It is the doctor's decision what tests to do. There is often a protocol, which prescribes what tests to perform depending on the preliminary diagnosis. The tests can be executed in any order or sometimes concurrently.

Therefore the so-called "repeater pattern" has been introduced and each test that can be selected is given as a transition bordered workflow, here represented as "test 1" till "test 3". In fact we assume that the token that enters in "start" has a value that is a set of identities of the tests to be executed. Each test workflow has an initial transition that with as precondition the identity of one test.

In the repeater pattern transition "decomposer" selects an arbitrary element of the test set in the token of place "a", puts the selected item in a token in place "b" and returns the set without the selected element to a token in place "a". Transition "check for empty set" fires only if the set of tests is empty and puts a token with the empty set as value in place "c". The transition "composer" will collect all test results from place "d" and collects them in the token in place "c". If all tests are collected transition "stop" can fire.

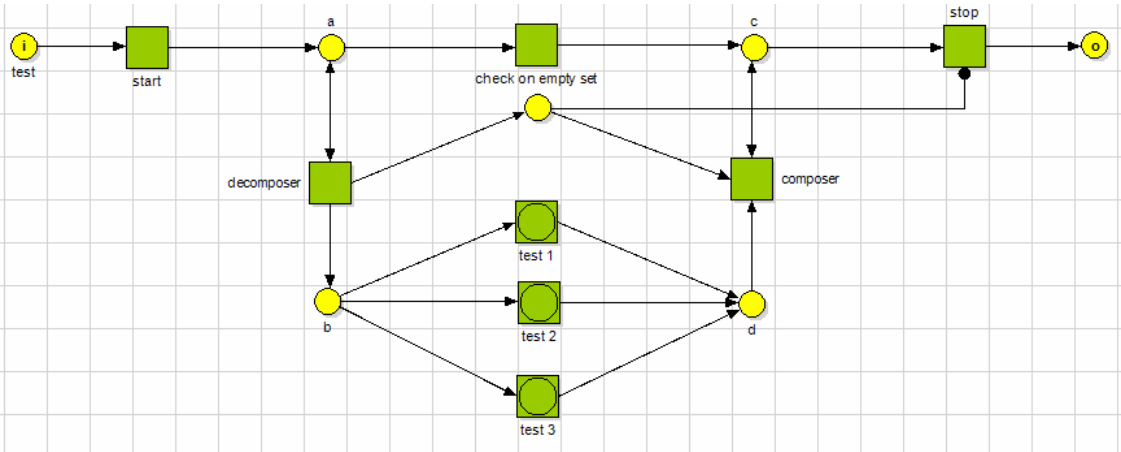

Figure 6-6 testing

Therapy

It is the doctor's decision what to do. In case of surgery or radiotherapy another specialist is involved who may consider the therapy as a new case and she might start with a more specific diagnosis process before the therapy will start.

Since there may be many therapies, that can be executed simultaneously, we have applied a repeater pattern as in "testing". The difference with that construction is that we do not use inhibitor arcs, but that we have limited the number of therapies that can be executed (100 here).

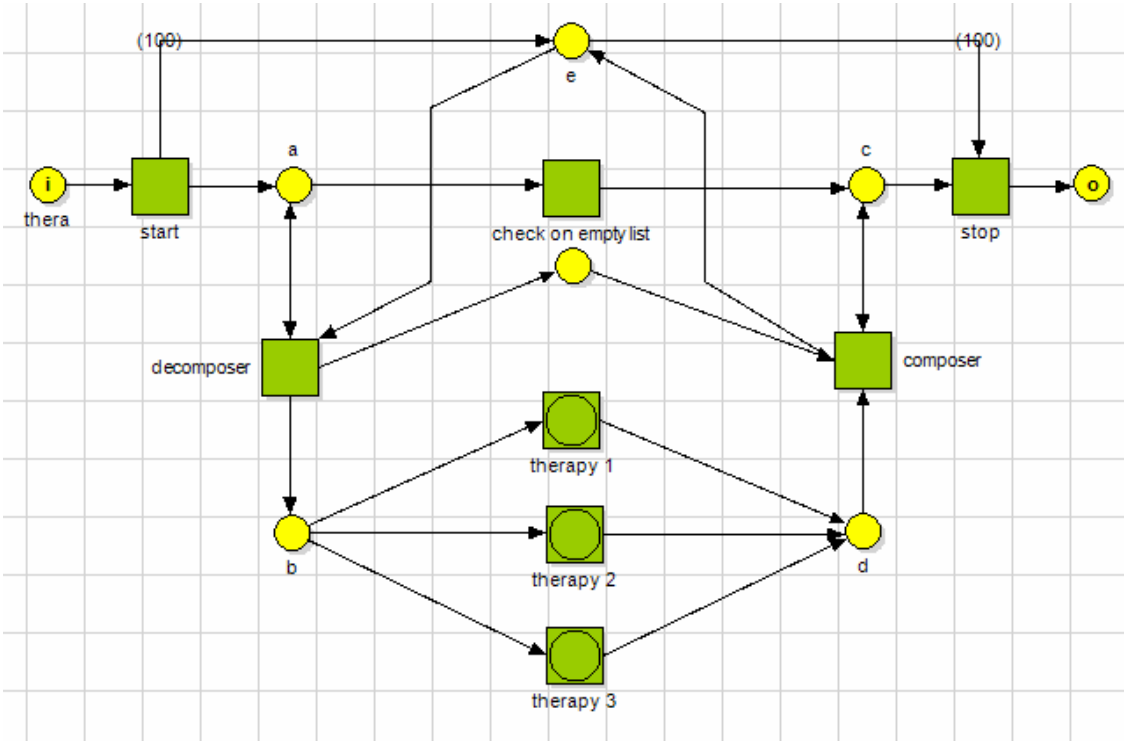

Figure 6-7 therapy

6.3 Process definition

The Infopath-Yasper process definition is consists of two parts: the Petri net definition and the Infopath solution.

6.3.1 Petri Net definition

Due to the fact that not all the functionalities of Yasper are used in Infopath-Yasper, and subnets and the patterns used in the case description cannot be implemented. Instead, another Petri net as show in figure 6-8 is defined to capture the behavior model of the process.

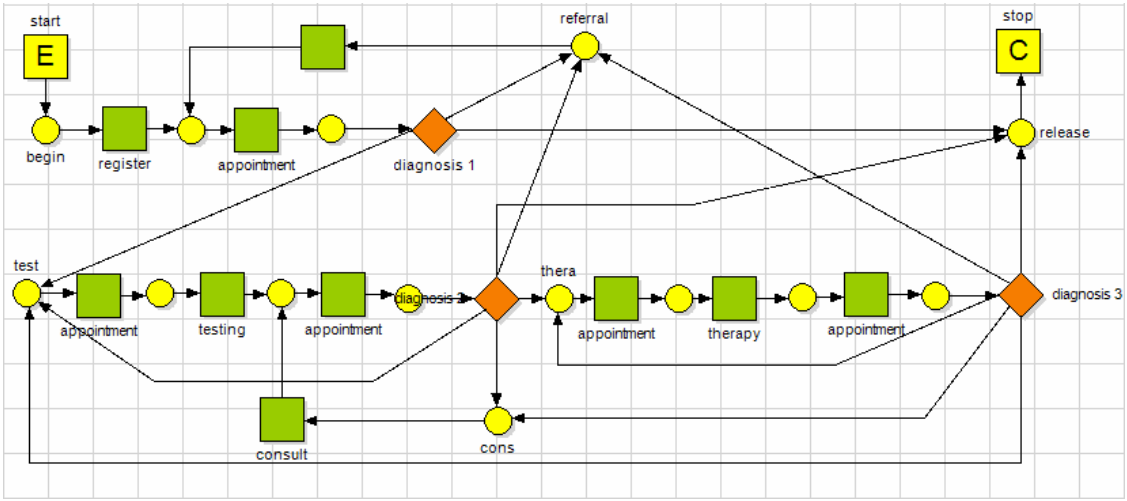

Figure 6-8 implementation in Yasper

Roles and views are assigned to the transitions that need human actor to perform. For instance transition *register* should be performed by *Receptionist*, and when the *receptionist* executes the task, view *Register* will be applied. The *description* field of the transition is used to define the name of the view (figure 6-9).

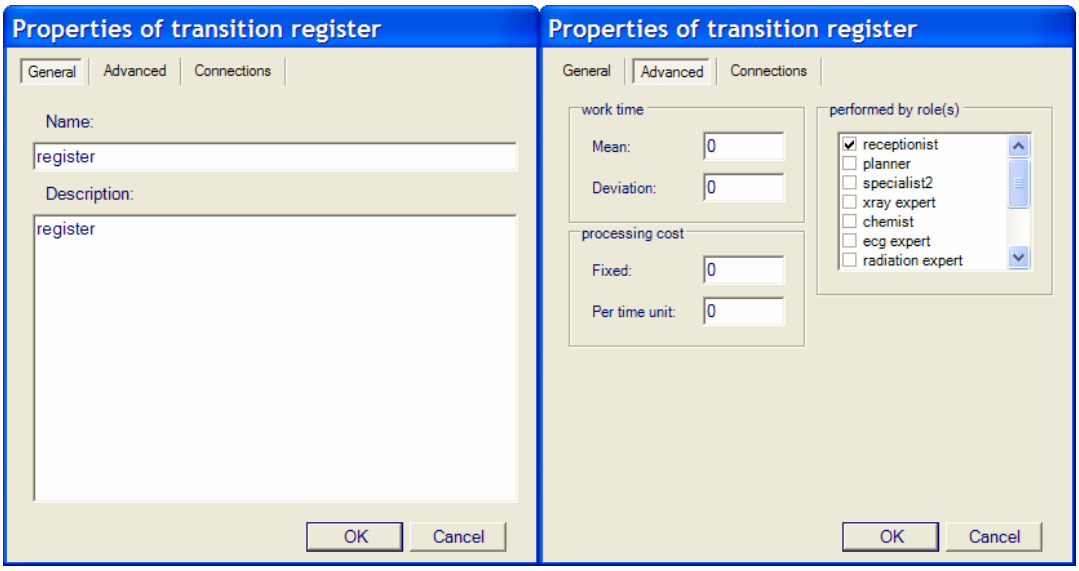

Figure 6-9 properties of transition

Table 6-1 shows the relation among transitions, roles and views.

Transition	**Role**	**View**
Register	Receptionist	Register
Appointment	Schedule planner	Appointment
Diagnosis 1	Doctor	First diagnosis
Diagnosis 2	Doctor	Second diagnosis
Diagnosis 3	Doctor	Third diagnosis
Referral	Doctor	referral
Test	Test expert	Test
Therapy	therapist	therapy

Table 6-1 transition-role-view

6.3.2 InfoPath solution (Electronic Patient Record)

The Infopath solution defines the data flow of the process, which is stored as an Electronic Patient Record. It contains the basic information of the patient such as full name, gender, date of birth and so on, as well their contact information and insurance policy data. It should include the patient's history record. Each diagnosis, test result, and therapy report should be record and retrievable for later usage.

This information model can be used as the electronic patient record.

Data connection

A data connection is build to submit the form to the workflow engine. There are four field that need to be set: the xml document, the Run Instance ID, the Task ID, and the choice. The *xmldoc* indicates the form itself. The *RunInstanceID* and *TaskID* are used to identify the forms, so they are mandatory. The *Choice* only needs to be filled when the transition is a XOR-split, meaning that the use has to indicate what the next transition should be.

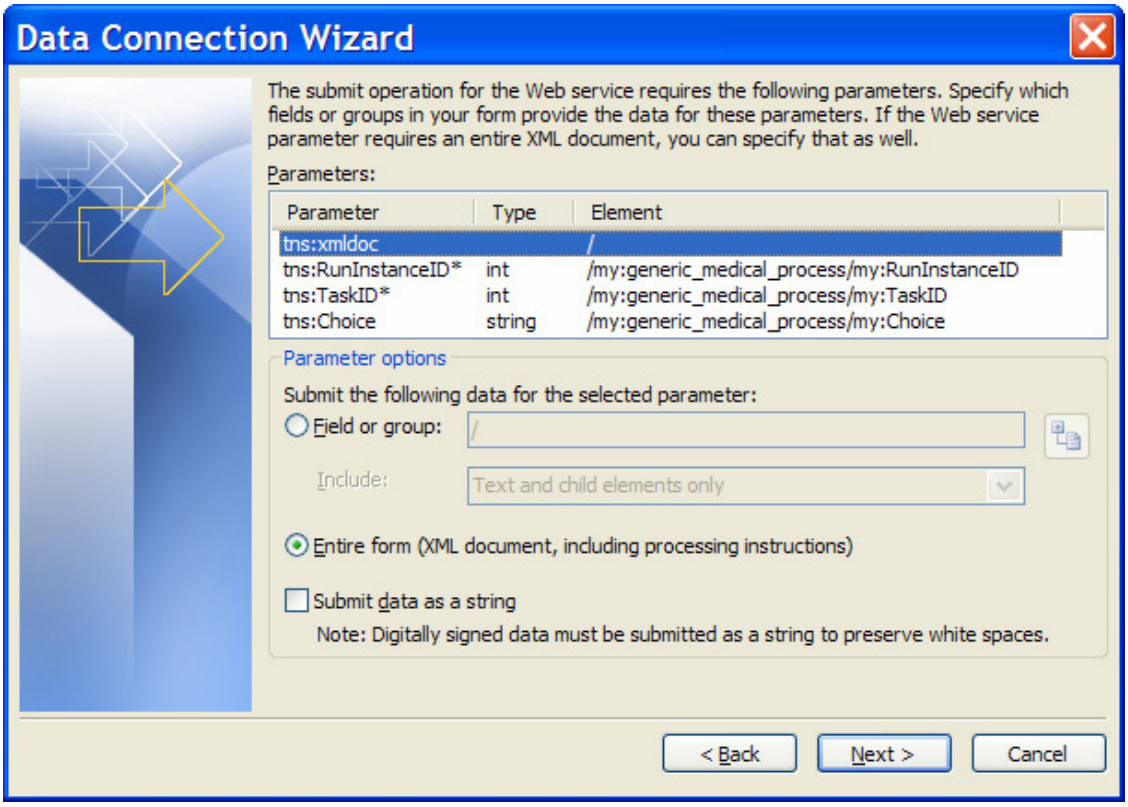

Figure 6-10 data connection

Data sources

Data structure

A patient should have one set of personal data and one set of contact data. For every visit of the patient will have at last one appointment record, and at least one diagnosis. A diagnosis is followed by a referral record, a consult record, test results, or treatment reports. Figure 6–11 shows the relation of the data.

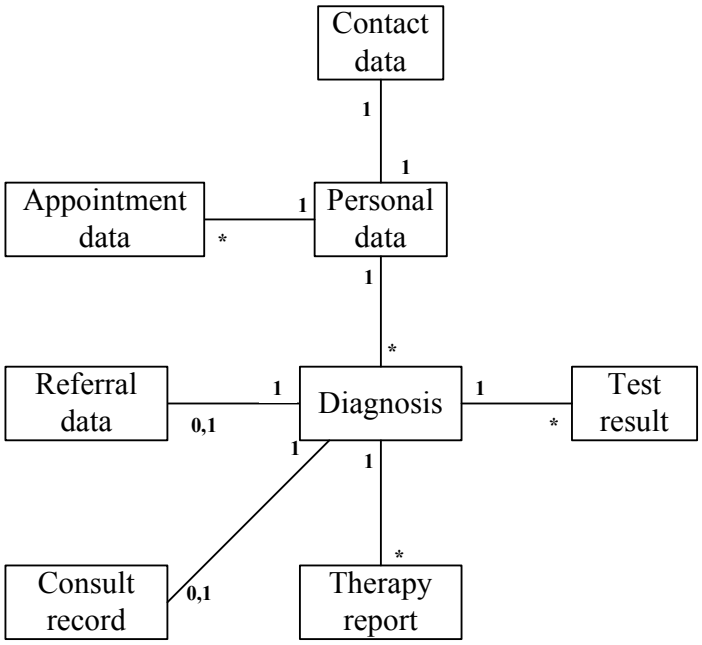

Figure 6-11 data structure of GMP

Data fields in Infopath are organized by groups, for example the personal_data group consists fields: FirstName (string), Lastname (string), Gender (string), DateOfBirth (date). For each form this group one occurs once, so the group is not repeatable. Groups such as appointment that can occur more than one times are set to repeatable groups. Figure 6-12 shows the implementation of the data source in Infopath.

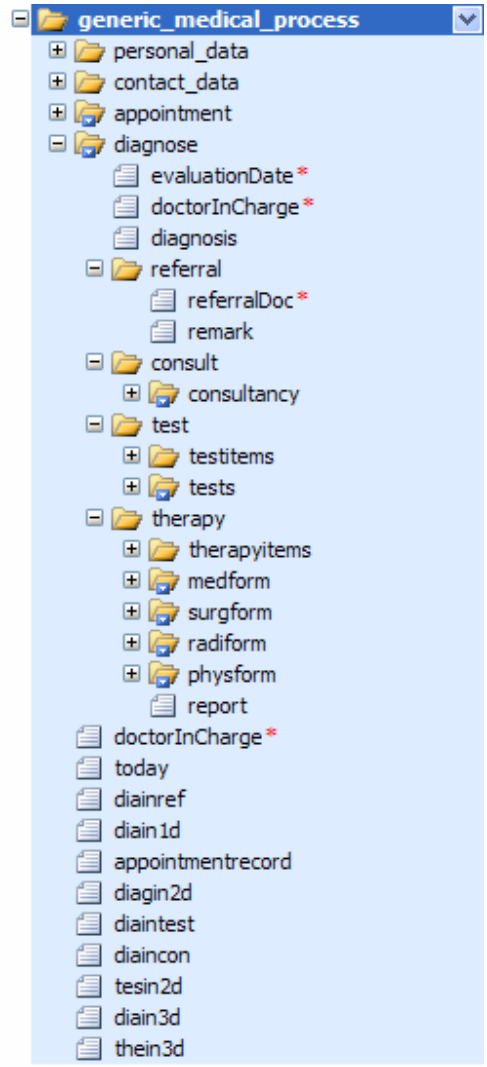

Figure 6-12 data source

There are several rules and constrains are applied to the data source:

- Mandatory constrain

  Some information is really crucial to the process, so they can not be empty, for instance the name and date of birth are used to identify the patient, therefore mandatory constrain are put on those fields so while fills out the from, the user will get alert when those fields are still empty.

- Logical constrain

To avoid small incautious mistakes, we can add some constrains to the model, such as: the for each medicine prescription the *starting date* should be earlier than the *ending date*.

- Format constrain

In order to get accurate and unified information some field are restrict to certain format, for instance the *email* filed should start with a string followed by a @ and then followed by at least one dot.

Views

Views are defined to present the xml document to different users and for different tasks.

In each view, the form can only present parts of the data source that are relevant to a certain task and certain users, the rest of the contain of the XML document is hided and protected. In addition, the designer can decide the layout and the format of the presentation.

There are several rules and constrains are applied to the view and controls:

- Read-only and disable constrain

Some information should not be changed at all during the process, such as the name of the patient or the test result, and some information can only be altered by certain people such as the diagnosis can only be edit by the doctor, so we set the status of the fields to read-only or the status of the controls to disable so that they can not be edit any more.

- Hide and filter

It is neither efficient nor secure to show old the history information, so some hide the information from the user, or filter the data with certain condition.

- Rule

Rules can be defined to controls to perform certain actions such as switch views.

Figure 6-13 illustrates an example view: first diagnosis.

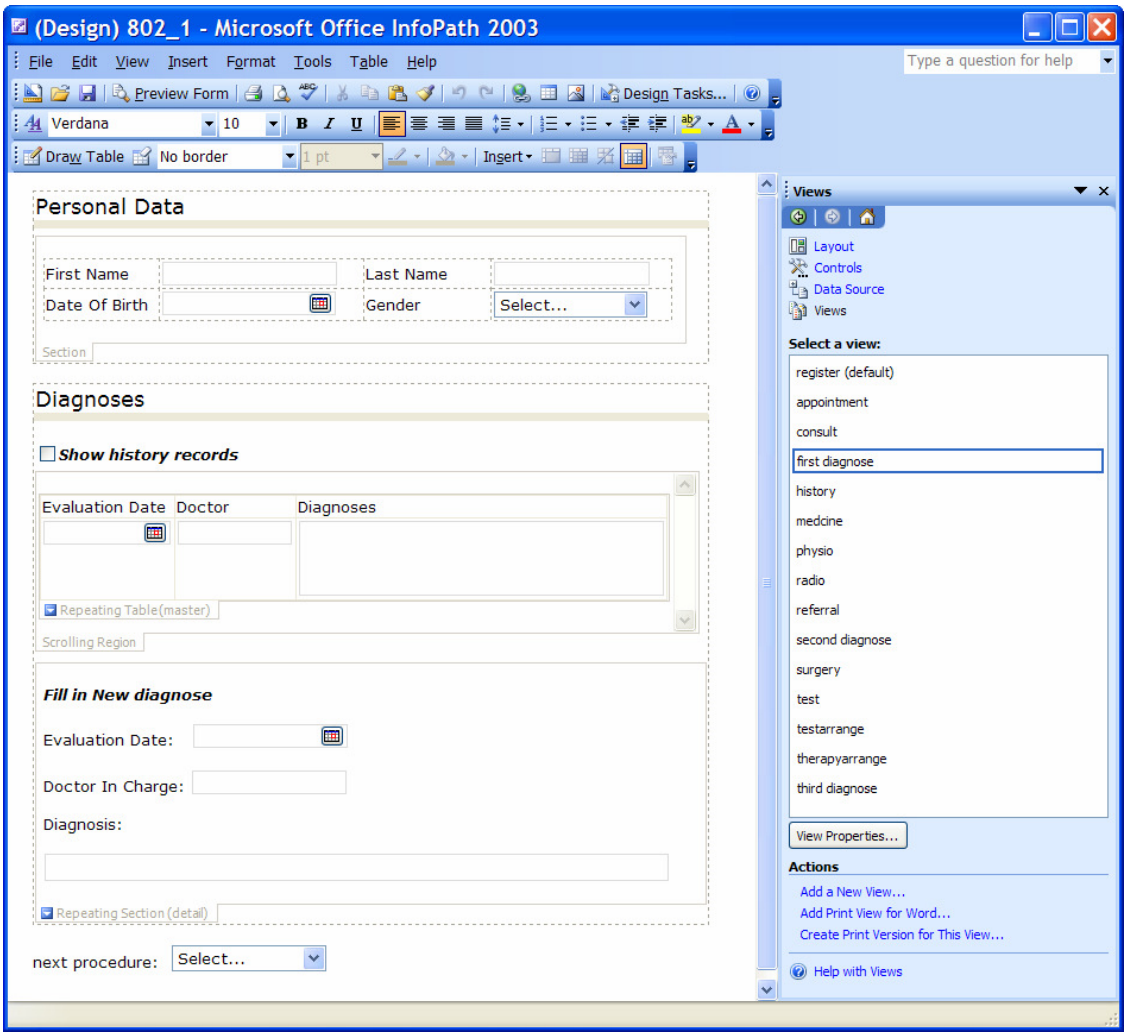

Figure 6-13 View: first diagnosis

The views that are defined for the generic medical process are:

- Register
- Appointment
- First diagnosis
- Second diagnosis
- Third diagnosis
- referral
- Test
- Radio
- Surgery
- Physio
- Medicine

6.4 Project administration

Figure 6-13 illustrate how the project manager should create the project type.

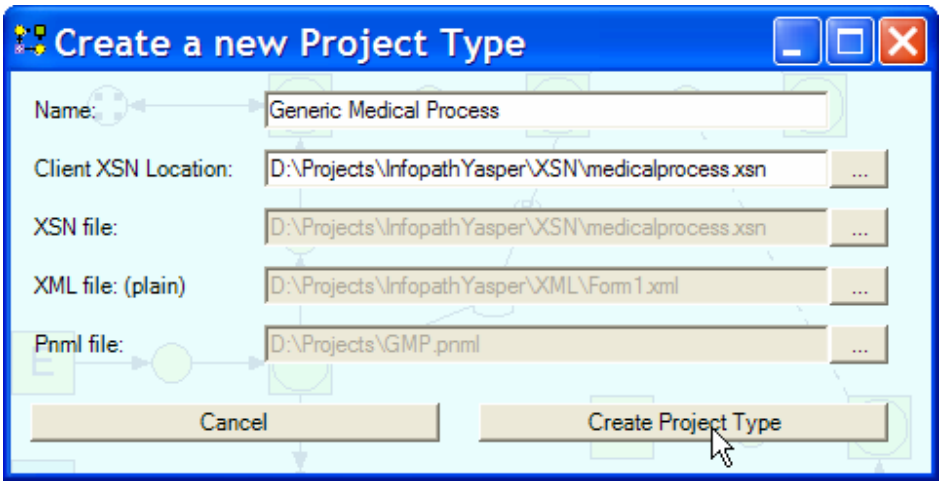

Figure 6-14 Admin Project control

After project type *Generic Medical Process* is created, the manager can start a project instance of that project type.

6.5 End user-usage

Figure 6-13 shows a screenshot made during the use of an end user.

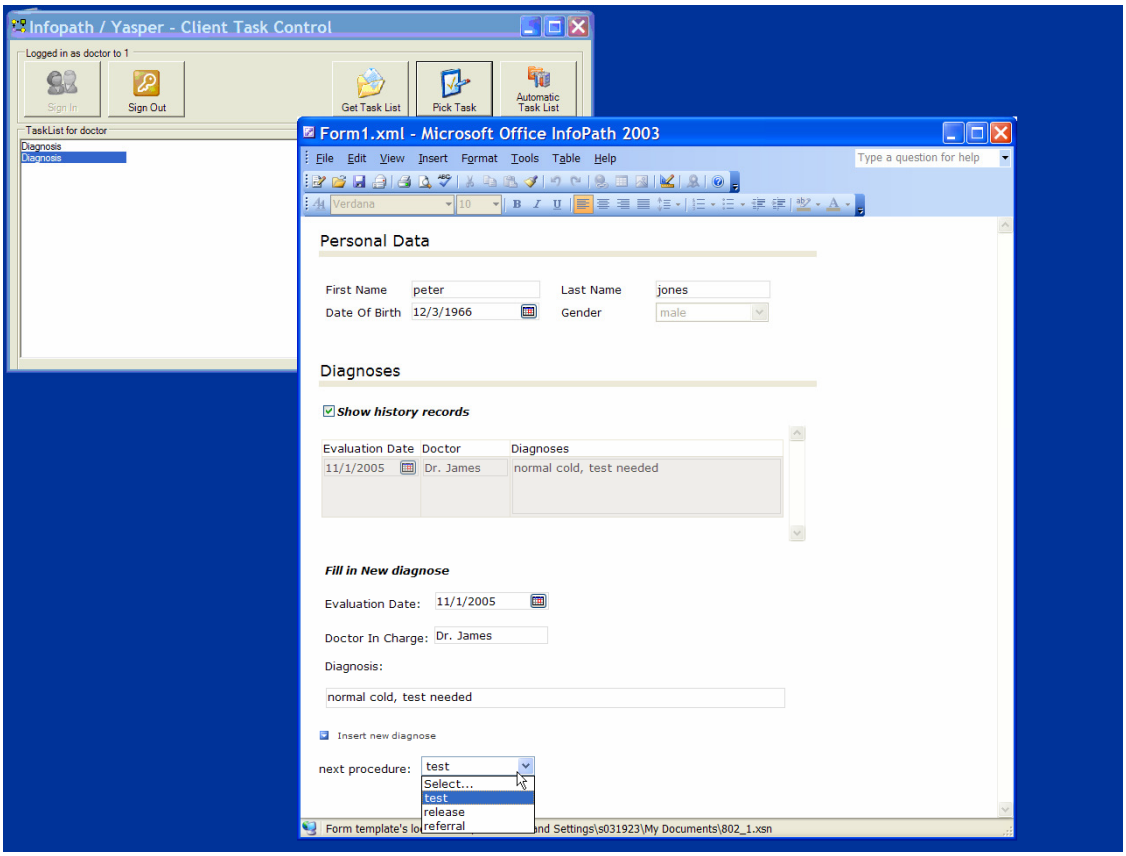

Figure 6-15 screenshot of end user interface

6.6 Evaluation of the models

Strong points

The process is modeled from the case's point of view and tasks are distributed to individual roles like the doctors and nurses, who only need to mind their own tasks of the process.

The cases are navigated through the organization, and the next step is always determined.

Due to data validation function provided by Infopath, the quality of users' input are assured.

The system automatically produced an Electronic Patient Record for each patient. The records can be used for the reference of diagnosis, medical research or disease management use. The Electronic Patient Record is stored as an XML document that can be easily exchanged through networks, and read by other systems.

Weak points

It is difficult to supervise and evaluate the performances of the employees, because they log in the system as a role rather than individuals. Some future work needs to be done to build a personal management function to the system.

The process is modeled based on some general rules such as: what procedure should the patient follows and who should perform what task, however there are more specific roles and guidelines that need to be specified. In General, Infopath-Yasper defined the behavior model and the information model of the generic medical process, in addition a rule base should also be defined explicitly to support the decision making process.

Infopath-Yasper does not provide any statistic analysis function, although the XML file can be analyzed by other querying tools.

The security issue is not addressed in the implementation. Some sensitive information should be protected for the sake of doctor-patient privacy. User should be coupled to a password. The communication between YasperWE and the client-side applications should be secured.

7 Conclusion and future development

Conclusion

In resent years, WfMSs are widely used in all kinds of organizations to support execution of business processes. Compared to other Enterprise Information System such as ERP and CRM, WfMSs focus on manage the execution of a process, while ERP systems emphasis organizing the resources along the supply chain, and CRM systems emphasis maintaining the information of the client relationship and the communication. Nevertheless, it should be noticed that the data flow of a process is as important as the control flow in a WfMS. A case management system is defined in this paper as a type of WfMSs that define, create and manage the execution of workflows, as well as manage the internal and external information related to the workflows. The difference between a normal a WfMS and a case management system is that a case management system has an internal information management tool to maintain the case related information.

With the work of our project, the research objectives are achieved:

1. *Define the main characteristics of a Case management system*

   A case management system should be flexible, information oriented, and it should support execution time decision, data validation, and information analysis.

2. *Find out what is missing in Regas*

   The analysis on Regas shows that as a case management system, the control flow definition and management part of Regas is weak, and a more expressive control flow definition component and a workflow engine are missing from its structure.

3. *Find out a solution to improve Regas*

   Yasper should be integrated with Regas_Admin to define the control flow of the processes. YasperWE should be built on the server-side to manage the execution of the workflow according to the process definition.

4. *Develop a Case management system*

   A case management system called Infopath-Yasper is developed. The system is designed based on two existing software tools: Infopath and Yasper, using XML and Web service technology. It manages the control flow of the processes as well as the data flow.

Besides, a case study on a Generic medical process is implemented with Infopath-Yasper. The result proved that this system is able to specify this kind of processes, control the workflow, and organize the resources. The system supports the organizations to achieve its business objectives.

The implementation also suggests a solution to handle hospital processes, and a method to produce automatic EPR. We believed this case study plays a positive role in medical process management area.

Future work

Future work for Regas:

The alternative solution for Regas should be realized based on the development of Infopath-Yasper and the implementation of YasperWE.

Future work for Infopath-Yasper:

- The merging of XML file function should be realized.
- A rule engine should be developed for the system.
- Security should be implemented to protect the information and the system.

Reference

Reference

[AH02] W. van der Aalst, K. van Hee: workflow management, Models, Methods and systems, Mass. MIT Press, Cambridge, 2002

[GH98] Volker Grhun, Monika Schneider: *workflow management base don process model repositorues,* IEEE 1998

[Davenport93] Davenport, T.H. *Process Innovation*, Harvard Business School Press, Boston, MA. 1993

[WFMC99] Workflow Management Coalition : *Terminology & Glossary* , workflow management coalition, Winchester, 1999

[LM04] Manuel Laguna, Johan Marklund: *Business process modeling, simulation, and* design, Pearson Prentice Hall 2004

[HLUPIC03] Vlaka Hlupic: *Knowledge and business process management*, London: Idea group, 2003

[PG03] Giorgos Papavassiliou, Gregoris mentzas: *knowledge modeling in weakly-structured business process,* Journal of knowledge management VOL.7 No.2003

[SOSF04] Shazia Sadiq, Maria Orlowska, Wasim Sadiq, Cameron Foulger: *Data Flow and Validation in Workflow Modelling,* ADC 2004

[AL96] Glenn Ammons James R. Larust: *Improving Data-flow Analysis with Path Profiles*, Microsoft Research 1996

[BKV96] E. Berrebi, P. Kission, S. Vernalde, S. De Troch, J.C. Herluison, J. Fr_ehel, A.A. Jerraya, I. Bolsens*: Combined Control Flow Dominated and Data Flow Dominated High-Level Synthesis*, ACM, Inc. 1996

[AALST96] W.M.P. van der Aalst: Three good reasons for using a petri-net-based workflow management system, Camebridge, Massachusetts, Nov 1996.

[REIJERS02] Hajo Alexander Reijers: *Design and Control of Workflow Processes-Business Process Management for the Service Industry,* Technische Universiteit Eindhoven 2002

[Davenport96] Davenport, T.H. *Why reengineering failed: the fad that forgot people*, Fast Company,January, pp. 69-74. 1996

[OASIS03] *OASIS universal description, discovery and integration (UDDI),* http://uddi.org/pubs/uddi_v3.htmO; October 2003.

[W3C01] *W3C Web Services Description Language (WSDL) 1.1,* http://www.w3.org/TR/2001/NOTE-wsdl-20010315O; March 2001.

[W3C03] *SOAP Version 1.2 Part 1: Messaging framework,* http://www.w3.org/TR/2003/REC-soap12-part1-20030624/O; June

2003.

[PC05] C. Geuer-Pollmann, J. Claessens, *Web services and web service security standards*, Information Security Technical Report 10, 2005.

[BF05] K. Beznosov, D. J. Flinn etc. *Introduction to Web services and their security*, Information Security Technical Report, 2005.

[PBTWOZ04] Mor Peleg, Aziz Boxwala, Samson Tu, Dongwen Wang, Omolola Ogunyemi, Qing Zeng . *Guideline Interchange Format 3.5 Technical Specification*, InterMed Collaboratory ,2004 may

[WS04] H.Wortmann, N.Szirbik, *Enterprise Information System*, TU/e, 2004

[CP03] I.J. Chen, K.Popovich, *Understanding customer relationship management (CRM),* Cleveland State University, USA, 2003

[White05] J.W. White, *Making ERP work the way your business works,* Fuego ERP white paper, 2005

[Siebel05] *Customer Relationship Management and Business Intelligence Applications,* http://www.siebel.com/