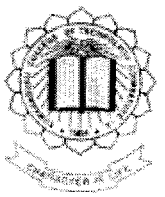


WorkFlow Mining for Discovering Process Models from Event Logs

By

Santhamani. R
Reg. No. 71204621034

Of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE

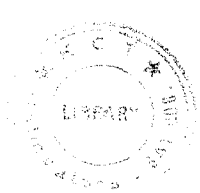
A PROJECT REPORT
Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree
of

MASTER OF COMPUTER APPLICATIONS

July, 2007



Certificate

Kumaraguru College of Technology

Coimbatore – 641006.

Department of Computer Applications

Bonafide Certificate

Certified that this project report titled **WorkFlow Mining for Discovering Process Models from Event Logs** is the bonafide work of **Ms. Santhamani R** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Project Guide

Head of the Department

Submitted for the University Examination held on 03-07-2007

Internal Examiner

External Examiner 3/7/07

Company Certificate

Softji Solutions
3A/11, K.T.R Nagar, Kolathupalayam, Thondamuthur (PO), Coimbatore - 641 109. Mobile : 98650 - 25454

25.6.2007

To whomsoever it may concern

This is to certify that **Ms. SanthaMani R (04MCA34),** final year student of Kumaraguru College of Technology, Coimbatore, completed the project titled "**WorkFlow Mining for Discovering Process Models from Event Logs** " for our organization towards partial fulfillment of the degree of MCA.

The duration of the project was from 18 Dec, 2006 to 18 May, 2007; his performance during the period was good.

We wish him success in his future endeavors.

For SOFTJI
Proprietor

Abstract

Abstract

Project management is a proven methodology for defining projects, scheduling projects, and tracking project progress against goals. The fundamental characteristics of Project Management are clear objectives and task definition, detailed planning, detailed scheduling of tasks commitments of people, equipment, and resources (time and money), management commitment to project support, continuous tracking, updating, and review of schedule, and constraints, continuous communication with team members and managers. Project management is not reorganization into a "full" project company or matrix structure only for technical or engineering activities; but an entirely new set of skills and behaviors unlike, anything done in "normal" management.

Contemporary workflow management systems are driven by explicit process models, i.e., a completely specified workflow design is required in order to enact a given workflow process. Creating a workflow design is a complicated time-consuming process and, typically, there are discrepancies between the actual workflow processes and the processes as perceived by the management. The starting point for such techniques is a so-called workflow-log, which is generated during project management. It contains information about the workflow process as it is actually being executed. In our project, we extract a process model from such a log and represent it in terms of a Petri net. We start with the information gathered about the workflow processes as they take place.

Acknowledgement

Acknowledgement

I express my grateful thanks to our beloved principal, **Dr. Joseph V Thanikal** and our former principal **Dr. K.K.Padmanabhan**, Kumaraguru College of Technology, Coimbatore, for giving me an opportunity to take up this project.

I express my deep sense of gratitude to **Dr.S.Thangasamy**, Professor and Dean of Computer Science and Engineering department and **Dr.M.Gururajan,** Professor and Head of Department of Computer Applications for extending their help in providing all the facilities at college for the successful completion of the project.

I would like to express my deepest and sincere gratitude to **Mr.A.Muthukumar**, Assistant Professor and Course Co-ordinator, of the Department of Computer Applications, for his guidance, support, cooperation and valuable suggestions during the course of this project.

I also thank our external guide **Mr. M.G.Jai Venkatesh,** SoftJi Solutions, Coimbatore, for providing me with adequate technical support and for his excellent guidance during the course of the project.

Contents

Table of Contents

List of Tables

List of Figures

Introduction

CHAPTER 1

INTRODUCTION

1.1 SYSTEM OVERVIEW

During the last decade, workflow management concepts and technology have been applied in many enterprise information systems. Workflow management systems such as Staffware, IBM MQSeries, COSA, etc., offer generic modeling and enactment capabilities for structured business processes. By making graphical process definitions, i.e., models describing the life-cycle of a typical case (workflow instance) in isolation, one can configure these systems to support business processes. Besides pure workflow management systems, many other software systems have adopted workflow technology. Consider, for example, ERP (Enterprise Resource Planning) systems such as SAP, PeopleSoft, Baan and Oracle, CRM (Customer Relationship Management) software, etc. Despite its promise, many problems are encountered when applying workflow technology. One of the problems is that these systems require a workflow design, i.e., a designer has to construct a detailed model accurately describing the routing of work. Modeling a workflow is far from trivial: It requires deep knowledge of the workflow language and lengthy discussions with the workers and management involved.

Instead of starting with a workflow design, we start by gathering information about the workflow processes as they take place. We assume that it is possible to record events such that

- Each event refers to a task (i.e., a well-defined step in the workflow),
- Each event refers to a case (i.e., a workflow instance), and

- Events are totally ordered (i.e., in the log events are recorded sequentially, even though tasks may be executed in parallel).

Any information system using transactional systems such as ERP, CRM, or workflow management systems will offer this information in some form. Note that we do not assume the presence of a workflow management system. The only assumption we make is that it is possible to collect workflow logs with event data. These workflow logs are used to construct a process specification, which adequately models the behavior registered.

1.2 COMPANY PROFILE

SoftJi Solutions is fast growing information technology in Coimbatore. It started with 50 skilled software persons and now it has strength of 140 Software and 20 Networking Hardware Engineers.

The core product of the company is a Web Server side page in .Net technologies. It mainly deals with the development of New Internet applications, New Intranet applications, Web enable existing legacy applications and Expertise.

SoftJi solutions technical expertise can be broadly classified as below. However, each group is not a water- tight compartment.

1. MS SQL Server, MS Access, Oracle Database design, administration, optimization, stored procedures, triggers and System Controlling Software - C, C++, and Java.
2. Power builder, Visual Basic development with high end user interfaces designs.
3. Tuning existing applications for faster results
4. Project specification and planning
5. Web development/net commerce (B2B, B2C) – HTML, DHTML, ASP, ASP.NET
6. Providing training, mentoring services, Hardware and Software recommendations.

System Study and Analysis

CHAPTER 2

SYSTEM STUDY AND ANALYSIS

2.1 PROBLEM STATEMENT

Now-a-days enormously amount of data are accumulated in all places like departmental stores, shops, hospital, school, college , all type of industries and also in Web. Web contains vast collection of document. It provides answer for almost all the query from variety of user. After the advent of computer the data are enormously available and by making use of such raw collection data to invent the knowledge is the process of Data Mining. Like wise in Web also plenty of Web Documents resides in online. Web is repository of variety of information like Technology, Science, History, Geography, Sports Politics and others. Similar way the routine jobs or work in any industry can be formulated and data store can be produced. From this data, we can apply data mining tool or algorithm to generate the knowledge through which Process Model can be formed. The starting point for such techniques is a so-called workflow-log, which is generated during project management. It contains information about the workflow process as it is actually being executed. In our project, we extract a process model from such a log and represent it in terms of a Petri net. We start with the information gathered about the workflow processes as they take place.

2.2 EXISTING SYSTEM

The Data mining Algorithms can be categorized into the following :

- Association Algorithm
- Classification
- Clustering Algorithm

2.2.1 Association

Association rule are used to discover elements that co-occur frequently within a data set consisting of multiple independent selections of elements (such as purchasing transactions), and to discover rules, such as implication or correlation, which relate co-occurring elements. Questions such as "if a customer purchases product A, how likely is he to purchase product B?" and "What products will a customer buy if he buys products C and D?" are answered by association-finding algorithms. This application of association rule learners is also known as market basket analysis. As with most data mining techniques, the task is to reduce a potentially huge amount of information to a small, understandable set of statistically supported statements.

In text books as well as in the business literature, market basket analysis is often promoted as a means to obtain product associations to base a retailer's promotion strategy on. They argue that associated products with a high lift/interest can be promoted effectively by only discounting just one of the two products. Implicitly, they argue that market basket analysis automatically identifies complements. Academics, however, have shown that one should be careful with this conclusion. They show that this implicit assumption does not hold. Their empirical analysis reveals that market basket analysis identifies as many substitutes as complements. Therefore, market basket analysis should not be used to build a promotion expert system for retailers, unless supplemented by other, more empirical, methods of product relationship determination.

2.2.2 Classification

The process of dividing a dataset into mutually exclusive groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from one another, where distance is measured with respect to specific variable(s) you are trying to predict. For example, a typical classification problem is to divide a database of companies into groups that are as homogeneous as possible with respect to a creditworthiness variable with values "Good" and "Bad."

2.2.3 Clustering

The process of dividing a dataset into mutually exclusive groups such that the members of each group are as "close" as possible to one another, and different groups are as "far" as possible from one another, where distance is measured with respect to all available variables.

Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

- **Automated prediction of trends and behaviors**. Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data — quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

- **Automated discovery of previously unknown patterns**. Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales

data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

2.3 PROPOSED SYSTEM

Here we are going to implement mining algorithm to work log file. The work log is generated for Software Development company for the process of Project Management.

Project management is a proven methodology for defining projects, scheduling projects, and tracking project progress against goals. The fundamental characteristics of Project Management are clear objectives and task definition, detailed planning, detailed scheduling of tasks commitments of people, equipment, and resources (time and money), management commitment to project support, continuous tracking, updating, and review of schedule, and constraints, continuous communication with team members and managers.

All the activities like Project assigned to company along with end date, modules in that project associated with Team Leader with end date, from team leader the same module may be split into task and assigned to team member with stimulated hour to complete the assignment are all stored in Data base. From this report is generated and the mining association rule is applied to extract the process model called Petri Net.

The step involved is:

- Project Management in Software development in implemented in VB .Net
- From data base related Report generated
- From this report, i.e., work log, association rule algorithm is implemented to extract Process Model.

We use the term process mining for the method of distilling a structured process description from a set of real executions.

To illustrate the principle of process mining, we consider the workflow log shown in Table 2.1. This log contains information about five cases (i.e., workflow instances). The log shows that for four cases (1, 2, 3, and 4), the tasks A, B, C, and D have been executed. For the fifth case, only three tasks are executed: tasks A, E, and D. Each case starts with the execution of A and ends with the execution of D. If task B is executed, then task C is also executed. However, for some cases, task C is executed before task B. Based on the information shown in Table 2.1 and by making some assumptions about the completeness of the log (i.e., assuming that the cases are representative and a sufficient large subset of possible behaviors is observed), we can deduce for example the process model shown in Fig. 3.1. The model is represented in terms of a Petri net. The Petri net starts with task A and finishes with task D. These tasks are represented by transitions. After executing A, there is a choice between either executing B and C in parallel, or just executing task E. To execute B and C in parallel, two non-observable tasks (AND-split and AND-join) have been added. These tasks have been added for routing purposes only and are not present in the workflow log. Note that we assume that two tasks are in parallel if they appear in any order. However, by distinguishing between start events and end events for tasks, it is possible to explicitly detect parallelism.

case identifier	task identifier
case 1	task A
case 2	task A
case 3	task A
case 3	task B
case 1	task B
case 1	task C
case 2	task C
case 4	task A
case 2	task B
case 2	task D
case 5	task A
case 4	task C
case 1	task D
case 3	task C
case 3	task D
case 4	task B
case 5	task E
case 5	task D
case 4	task D

Table 2.1: Work Flow Log

Start events and end events can also be used to indicate that tasks take time. However, to simplify the presentation, we assume tasks to be atomic without losing generality. In fact, in our tool EMiT, we refine this even further and assume a customizable transaction model for tasks involving events like "start task," "withdraw task," "resume task," "complete task," etc. Nevertheless, it is important to realize that such an approach only works if events like these are recorded at the time of their occurrence. The basic idea behind process mining, also referred to as workflow mining, is to construct Fig. 3.1 from the information given in Table 2.1. In this paper, we will present a new algorithm and prove its correctness.

2.4 FEASIBILITY ANALYSIS

Feasibility analysis is the measure of how beneficial or practical the development of Information System will be to the Organization. Once the problem is explained information is gathered about the system to test whether the system is viable Technically, Financially and Operationally. Thus, feasibility study is carried out in three phases as follows:

2.4.1 Technical Feasibility

Technical Feasibility is the measure of practicality of a specific technical solution and the availability of technical resources and expertise. It centers on the existing computer system (hardware, software, etc.) and to what extent it can support the new addition.

The proposed system is to be developed using .Net, which is one of the leading technologies in the market. These resources are easily available and the company does not need to acquire any development license. These features of the selected technology are quite beneficial to the proper functioning of the system in different environments.

2.4.2 Operational Feasibility

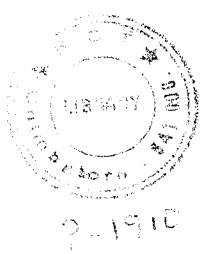
Operational Feasibility asks if the system will work when it is developed and installed. It checks for the support of the management, the current business methods, user's involvement and their attitude towards the proposed system, etc.

The proposed system has found encouraging support from the company staff and management as it will be of great use to them. The employees of the organization are also committed to have the system operational as it will save time and reduce their workload.

2.4.3 Economic Feasibility

Economic Feasibility is the measure of the cost-effectiveness of the proposed system. The investment to be made in the proposed system must prove a good investment to the organization by returning benefits equal to or exceeding the costs incurred in developing the system.

The proposed benefits of the system will outweigh the costs to be incurred during system developed since the system does not require procurement of additional hardware facilities it is economically feasible. In addition capability of the system to incorporate future enhancement will improve the performance to suit the future need of the concern.



Development Environment

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

The hardware support required for deploying the application:-

Processor	:	Intel Processor IV
RAM	:	128 MB
Hard disk	:	20 GB
CD drive	:	40 x Samsung
Floppy drive	:	1.44 MB
Monitor	:	15' Samtron color
Keyboard	:	108 mercury keyboard
Mouse	:	Logitech mouse

3.2 SOFTWARE REQUIREMENTS

The software support required for deployment is:-

Operating System – Windows XP/2000
Language used – VB .Net

3.3 PROGRAMMING ENVIRONMENT

3.3.1 Process Mining

Process mining is useful for at least two reasons. First of all, it could be used as a tool to find out how people and/or procedures really work. Consider, for example, processes supported by an ERP system like SAP (e.g., a procurement process). Such a system logs all transactions, but in many cases does not enforce a specific way of working. In such an environment, process mining could be used to gain insight in the actual process. Another example would be the flow of patients in a hospital. Note that in such an environment, all activities are logged, but information about the underlying process is typically missing.

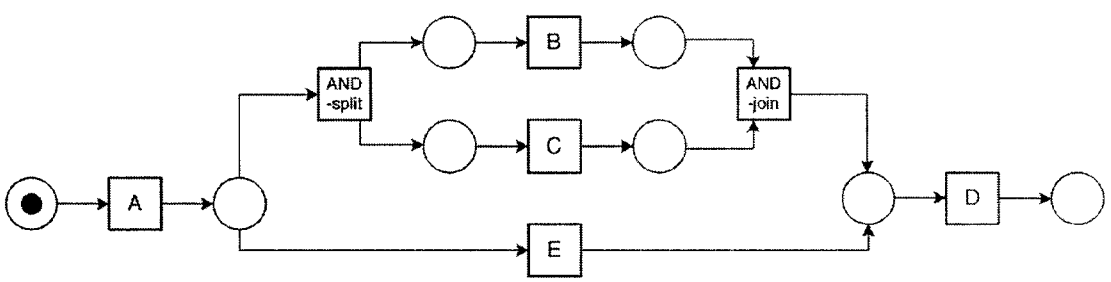


Fig 3.1 : Process Model Corresponding to Work flow log

In this context, it is important to stress that management information systems provide information about key performance indicators like resource utilization, flow times, and service levels, but not about the underlying business processes (e.g., causal relations, ordering of activities, etc.). Second, process mining could be used for Delta analysis, i.e., comparing the actual process with some predefined process. Note that in many situations, there is a descriptive or prescriptive process model. Such a model specifies how people and organizations are assumed/expected to work. By comparing the descriptive or

prescriptive process model with the discovered model, discrepancies between both can be detected and used to improve the process. Consider, for example, the so-called reference models in the context of SAP. These models describe how the system should be used. Using process mining, it is possible to verify whether this is the case. In fact, process mining could also be used to compare different departments/organizations using the same ERP system. An additional benefit of process mining is that information about the way people and/or procedures really work and differences between actual processes and predefined processes can be used to trigger Business Process Reengineering (BPR) efforts or to configure "process-aware information systems" (e.g., workflow, ERP, and CRM systems).

Table 2.1 contains the minimal information we assume to be present. In many applications, the workflow log contains a timestamp for each event and this information can be used to extract additional causality information. Moreover, we are also interested in the relation between attributes of the case and the actual route taken by a particular case. For example, when handling traffic violations: Is the make of a car relevant for the routing of the corresponding traffic violations? (For example, "People driving a Ferrari always pay their fines in time.")

For this simple example, it is quite easy to construct a process model that is able to regenerate the workflow log. For larger workflow models this is much more difficult. For example, if the model exhibits alternative and parallel routing, then the workflow log will typically not contain all possible combinations. Consider 10 tasks, which can be executed in parallel. The total number of interleaving is 10! = 3628800. It is not realistic that each interleaving is present in the log. Moreover, certain paths through the process model may have a low probability and, therefore, remain undetected. Noisy data (i.e., logs containing rare events, exceptions, and/or incorrectly recorded data) can further complicate matters. In this paper, we do not focus on issues such as noise. We assume that there is no noise and that the workflow log contains "sufficient" information.

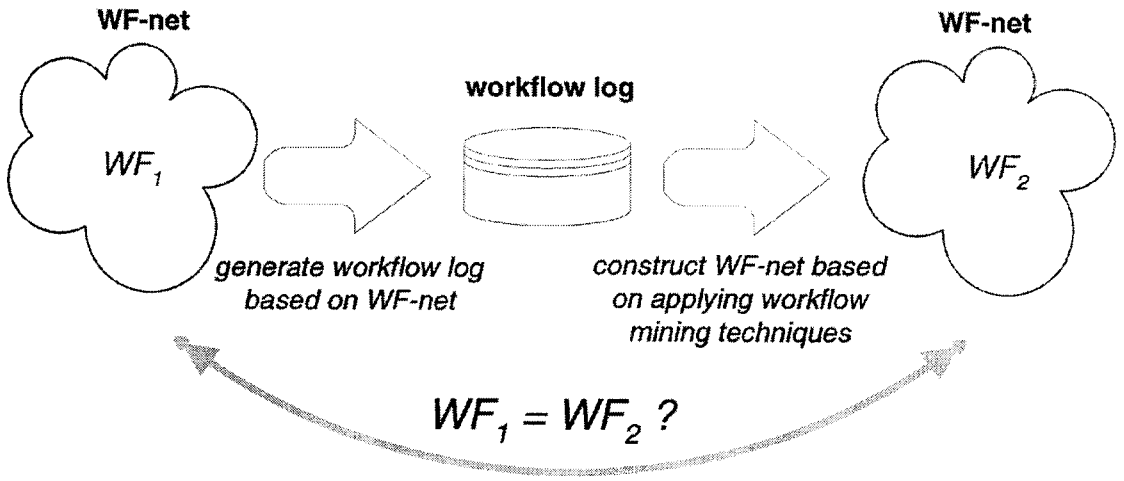


Fig 3.2: Workflow Net generated and designed with mining technique

Under these ideal circumstances, we investigate whether it is possible to rediscover the workflow process, i.e., for which class of workflow models is it possible to accurately construct the model by merely looking at their logs. This is not as simple as it seems. Consider, for example, the process model shown in Fig. 3.1. The corresponding workflow log shown in Table 2.1 does not show any information about the AND-split and the AND-join. Nevertheless, they are needed to accurately describe the process. These and other problems are addressed in this paper. For this purpose, we use workflow nets (WF-nets). WF-nets are a class of Petri nets specifically tailored toward workflow processes. Fig. 3.1 shows an example of a WF-net. To illustrate the rediscovery problem we use Fig. 3.2.

Suppose we have a log based on many executions of the process described by a WF-net WF1. Based on this workflow log and using a mining algorithm, we construct a WF-net WF2. An interesting question is whether WF1 ¼ WF2. In this paper, we explore the class of WF-nets for which WF1 ¼ WF2. Note that the rediscovery problem is only addressed to explore the theoretical limits of process mining and to test the algorithm presented in this project. We

have used these results to develop tools that can discover unknown processes and have successfully applied these tools to mine real processes.

The remainder of this project is organized as follows: First, we introduce some preliminaries, i.e., Petri nets and WF-nets. The quality of the algorithm is supported by the fact that it is able to rediscover a large class of workflow processes.

We use a variant of the classic Petri-net model, namely, Place/Transition nets.

An Place/Transition net, or simply P/T-net, is a tuple (P, T, F), where:

1. P is a finite set of places.

2. T is a finite set of transitions such that P n T = ? .

3. F sub set of (P X T) U (T X P) is a set of directed arcs, called the flow relation.

A marked P/T-net is a pair (N, s), where N = (P, T, F) is a P/T-net and where s is a bag over P denoting the marking of the net. The set of all marked P/T-nets is denoted N.

Most workflow systems offer standard building blocks such as the AND-split, AND-join, OR-split, and OR-join. These are used to model sequential, conditional, parallel, and iterative routing. Clearly, a Petrinet can be used to specify the routing of cases. Tasks are modeled by transitions and places and arcs model causal dependencies. In fact, a place corresponds to a condition, which can be used as pre and/or postcondition for tasks. An AND-split corresponds to a transition with two or more output places, and an AND-join corresponds to a transition with two or more input places. OR-splits/OR-joins correspond to places with multiple outgoing/ingoing arcs. Given the close relation

between tasks and transitions, we use the terms interchangeably. A Petrinet, which models the control-flow dimension of a workflow, is called a WorkFlow net (WF-net). It should be noted that a WF-net specifies the dynamic behavior of a single case in isolation.

3.3.2 Oracle 9i

Oracle Corporation strives to comply with industry-accepted standards and participates actively in SQL standards committees. The strengths of SQL provide benefits for all types of users, including application programmers, database administrators, managers, and end-users. Technically speaking, SQL is a data sublanguage. The purpose of SQL is to provide an interface to a relational database such as Oracle, and all SQL statements are instructions to the database.

Features of Oracle 9i

ORACLE 9i provides statements for a variety of tasks, including:

- Querying data
- Inserting, updating, and deleting rows in a table
- Creating, replacing, altering, and dropping objects
- Controlling access to the database and its objects
- Guaranteeing database consistency and integrity
- Supports PL/SQL

3.3.3 Visual Basic .Net

Microsoft Visual Basic .Net is used as a front-end tool. The reason for selecting Visual Basic .Net as front-end tool is as follows:

- Visual Basic .Net has flexibility, allowing one or more language to interoperate to provide the solution. This Cross Language Compatibility allows to do project at faster rate.
- Visual Basic .Net has Common Language Runtime, that allows the entire component to converge into one intermediate format and then can interact.
- Visual Basic .Net provides excellent security when your application is executed in the system
- Visual Basic .Net has flexibility, allowing us to configure the working environment to best suit our individual style. We can choose between a single and multiple document interfaces, and we can adjust the size and positioning of the various IDE elements.
- Visual Basic .Net has Intelligence feature that make the coding easy and also Dynamic help provides very less coding time.
- The working environment in Visual Basic .Net is often referred to as Integrated Development Environment because it integrates many different functions such as design, editing, compiling and debugging within a common environment. In most traditional development tools, each of separate program, each with its own interface.
- The Visual Basic .Net language is quite powerful – if we can imagine a programming task and accomplished using Visual Basic .Net.

- After creating a Visual Basic .Net application, if we want to distribute it to others we can freely distribute any application to anyone who uses Microsoft windows. We can distribute our applications on disk, on CDs, across networks, or over an intranet or the Internet.
- Toolbars provide quick access to commonly used commands in the programming environment. We click a button on the toolbar once to carry out the action represented by that button. By default, the standard toolbar is displayed when we start Visual Basic. Additional toolbars for editing, form design, and debugging can be toggled on or off from the toolbars command on the view menu.
- Many parts of Visual Basic are context sensitive. Context sensitive means we can get help on these parts directly without having to go through the help menu. For example, to get help on any keyword in the Visual Basic language, place the insertion point on that keyword in the code window and press F1.
- Visual Basic interprets our code as we enter it, catching and highlighting most syntax or spelling errors on the fly. It's almost like having an expert watching over our shoulder as we enter our code.

System Design and Development

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 ELEMENTS OF DESIGN

System Design is the most creative and challenging phase in the development of a software system. Design implies to a description of the final system and the process by which it is developed. The first step is to determine what input data is needed for the system and then to design a database that will meet the requirements of the proposed system. The next step is to determine what outputs are needed from the system and the format of the output to be produced.

During the design of the proposed system some areas where attention is required are:

- What are the inputs required and the outputs produced?
- How should the data be organized?
- What will be the processes involved in the system?
- How should the screen look?

The steps carried out in the design phase are as follows:

- Modular Design
- Input Design
- Output Design
- Database Design

4.1.1 Modular Design

It is always difficult for any System Development team to grasp a system without breaking it into several smaller systems. These smaller systems will be a part of the original system yet they will be independent in the sense that they will incorporate within them the major functionalities of the proposed system.

A software system is always divided into several subsystems, which make it easier to develop and perform tests on the whole system. The subsystems are known as the modules and the process of dividing an entire system into subsystems is known as Decomposition.

The workflow mining process to create the Process Model for Project Management job is categorized as follows:

- Projects Admin
- Workflow Admin
- Work log
- Workflow Mining
- Results and Analysis

4.1.1.1 Projects Admin

This module contains three forms namely, projects, team leader and team member. Projects form contains Project code, project name, Platform, duration, starting date, completion date. By selecting a particular project we could obtain all the details of the project in this form.

Team leader form contains Team Leader Code, Project Code, Leader Name, Address, Date of Birth and Gender.

Team member form contains Team member Code, Project Code, member Name, Address, Date of Birth and Gender. This module is used to maintain the employee details of the organization.

These forms also helps in editing and deleting the existing information. Project and Team member forms also have additional option to add new information of new projects and team members respectively.

4.1.1.2 Workflow Admin

The user can specify the module details, add each team member to specified modules and add task details. The task details are used to generate the workflow; it contains details like starting date, completion date, estimated hours, hours taken, task weightage, etc.

The objectives of Project Management are task definition, detailed planning, detailed scheduling of tasks, equipment, and resources (time and money), management of project support, continuous tracking, updating, and review of schedule, and constraints, continuous communication with team members and managers.

In This Module user gathering information about the workflow processes from that organization and record events such that

1. Each event refers to a task (i.e., a well-defined step in the workflow),
2. Each event refers to a case (i.e., a workflow instance), and
3. Events are totally ordered (i.e., in the log, events are recorded Sequentially, even though tasks may be executed in parallel).

Adding and Deleting the details of Employees and Tasks involved in a project is controlled by Administrater and also partially by the Employee.

4.1.1.3 Worklog

This module keeps track of the team member, status of every task involved in a particular project. Workflow-log is generated during project management. It contains information about the workflow process when it is actually being executed.

This Module displays the generated workflow log, which contains details of all the tasks of all the modules involved in the project, with the necessary details for workflow mining.

Project assigned to company along with end date, modules in that project associated with Team Leader with end date, from team leader the same module may be split into task and assigned to team member with stimulated hour to complete the assignment are all stored in Worklog. From this report is generated and the mining association rule is applied to extract the process model called Petri net.

The project status is calculated from differences of starting date and Ending date of the project. The goal of the workflow mining is to find a workflow model (e.g., a WF-net) on the basis of the workflow log. The Task Details reports are used to display the task allocated to the particular employee.

4.1.1.4 Workflow Mining

In this module, we generate two workflow nets WFnet1 and WFnet2. A Petri net, which models the control-flow dimension of a workflow, is called a Workflow net (WF-net). A WF-net specifies the dynamic behavior of a single case in isolation.

WFnet1 is generated from the generated work log. WFnet2 is the proposed work log and it displays the weighted backlog error, which is the total percentage of difference in estimated hours and hours taken in each task. It displays the project accuracy percentage.

It finally generates and displays the Petri nets for the two-workflow nets, which is the visual representation of two work logs. The process model is represented in terms of a Petri net. The Petri net starts with task A and finishes with task D. These tasks are represented by transitions. After executing A, there is a choice between either executing B and C in parallel, or just executing task E. To execute B and C in parallel, two no observable tasks (AND-split and AND-join) have been added. These tasks have been added for routing purposes only and are not present in the workflow log.

We use a variant of the classic Petri-net model, namely Place/Transition nets. A Place/Transition net, or simply P/T-net, is a tuple (P, T, F), where:

1. P is a finite set of places.
2. T is a finite set of transitions such that P n T = ? .
3. F sub set of (P X T) U (T X P) is a set of directed arcs, called the flow relation.

A marked P/T-net is a pair (N, s), where N = (P, T, F) is a P/T-net and where s is a bag over P denoting the marking of the net. The set of all marked P/T-nets is denoted N. Tasks are modeled by transitions and places and arcs model causal dependencies.

Process mining is used for two reasons. First of all, it could be used as a tool to find out how people and/or procedures really work. Second, process mining could be used for Delta analysis, i.e., comparing the actual process with some predefined process.

4.1.1.5 Results and Analysis

In this module, we can observe the various characteristics like backlog error, project accuracy, petri nets, etc. for different projects with different types of tasks.

4.1.2 Input Design

The input design is the process of converting the user-oriented inputs into computer-based format. Input data is the most common case of errors in data processing. Errors entered by data entry operators can control by input design.

Input design is a part of the overall system design, which requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify these errors. Input can be categorized as internal, external, operational, computerized and interactive. The analysis phase should consider the impact of the inputs on the system as a whole and on the other systems.

The main objectives considered during input design are nature of input, processing flexibility and thoroughness of validation rules. Handling of priorities within the input documents Screen design to ensure accuracy and efficiency of input Relationship with files. Careful design of the input also involves attention to error handling, controls, batching and validation procedures.

The goal of designing input data is to make sure that the automation is easy, logical and free from errors. The input design requirements such as user friendliness, consistent format and interactive dialogue, which provide users with timely help and correct messages, are given high priority.

Workflow management systems will offer this information in some form. we do not assume the presence of a workflow management system. The only assumption we make is that it is possible to collect workflow logs with event data. These workflow logs are used to construct a process specification, which adequately models the behavior registered.

4.1.3 Output Design

Output generally refers to the results and information that are generated by the system. For many end users, output is the main reason for developing the system and the basis on which they will evaluate the usefulness of the application. Most end-users will noticeably operate the information system or enter data through workstation, but they will use the output from the system.

When designing output, system analyst must accomplish the following: Determine what information to present, decide whether to display, print the information and select the output medium.

Reports are generated as output for the users to view and take printouts. Different reports are generated for different criteria. The reports present in the system are:

- Worklog Report
- Worklog Process Model
- Petrinet (Actual Executed Log)
- Petrinet (Standard Proposed Log)

4.1.4 Database Design

Database design deals with the table structure and organization. The purpose of the database is to enable easy access of information for the user. The general theme behind databases is to handle the information as an integrated one.

While designing a database, we have to make decisions regarding how best to take some system in the real world and model it in a database. This process consists of deciding which tables to create and what columns they will contain as well as the relationships between tables.

A database is an integrated collection of user related data stored with minimum redundancy, serves many users/applications quickly and efficiently.

A database system is basically a computerized record keeping system, i.e., it is a computerized system whose overall purpose is to maintain information and make that information available on demand. RDBMS is collections or inter related data and set of programs that allow several users to access and manipulate the data. Its main purpose is to provide users with an abstract view of the data, i.e. the system hides certain details of how the data is stored and maintained.

A database is a collection of inter-related data stored with minimum redundancy to serve many users quickly and efficiently. The general objective of database design is to make the data access easy, inexpensive and flexible to the user. An elegantly designed database can play a strong foundation for the whole system.

The details about the relevant data for the system are first identified. According to their relationship, tables are designed through the following method.

- The data type for each data item in the table is decided.
- The tables are then normalized.

The tables are normalized so that they can provide better response time, have data integrity, avoid redundancy and be secure.

The tables for the Result Analysis system have been Normalized up to the Second Normal Form (2NF).

4.2 TABLE STRUCTURE

4.2.1 Table Name: PROJECT_DETAILS

Primary key : projectcode

Foreign key : Platform

Field Name	Data Type	Remarks
ProCode	Varchar2(10)	Project Code
Name	Varchar2(20)	Project Name
Platform	Varchar2(20)	platform
Duration	Varchar2(10)	duration
StartD	Date	Starting Date
Com	Varchar2(5)	completed
ComD	Date	Completion Date

4.2.2 Table Name: TEAMLEADER _DETAILS

Primary key : Teamleadercode

Foreign key : Projectcode

Field Name	Data Type	Remarks
TLCode	Varchar2(10)	Team Leader Code
ProCode	Varchar2(20)	Project Code
Name	Varchar2(20)	Leader Name
Add	Varchar2(30)	Address
DOB	Date	Date Of Birth
Gen	Varchar2(8)	Gender

4.2.3 Table Name: TEAMMEMBER_DETAILS

Primary key : TeamMemberCode

Foreign key : ProjectCode

Field Name	Data Type	Remarks
TMCode	Varchar2(10)	Team Member Code
ProCode	Varchar2(20)	Project Code
Name	Varchar2(20)	Member Name
Add	Varchar2(30)	Address
DOB	Date	Date Of Birth
Gen	Varchar2(8)	Gender

4.2.4 Table name: MODULE_DETAILS

Primary key: ModuleCode

Foreign key: ProjectCode

Field Name	Data Type	Remarks
MCode	Varchar2(10)	Module Code
Des	Varchar2(10)	Description
ProCode	Varchar2(20)	Project Code
TLCode	Varchar2(20)	Team Leader Code
StartD	Date	Starting Date
EstD	Varchar2(8)	Estimation Date
Com	Varchar2(8)	Completed
ComD	Date	Completion Date

4.2.5 Table Name: TASK_DETAILS

Primary key: TaskCode

Foreign key: ProjectCode

Field Name	Data Type	Remarks
TCode	Varchar2(10)	Task Code
MCode	Varchar2(10)	Module Code
ProCode	Varchar2(20)	Project Code
TMCode	Varchar2(20)	Team Leader Code
Des	Varchar2(10)	Description
Weight	Varchar2(8)	Weight
Dval	Date	Date Value
EstHour	Varchar2(10)	Estimation Hours
Com	Varchar2(8)	Completed
HTaken	Varchar2(8)	Hours Taken

4.2.6 Table Name: REGISTERED_TEAMMEMBER

Primary key : TeamMemberCode

Foreign key : ModuleCode

Field Name	Data Type	Remarks
MCode	Varchar2(20)	Module Code
TMCode	Varchar2(20)	Team Member Code
RegD	Date	Registration Date

4.3 DATA FLOW DIAGRAMS

Data flow diagrams are graphical representation depicting information regarding the flow of control and the transformation of data from input to output. The DFD may be used to represent the system or software at any level of abstraction.

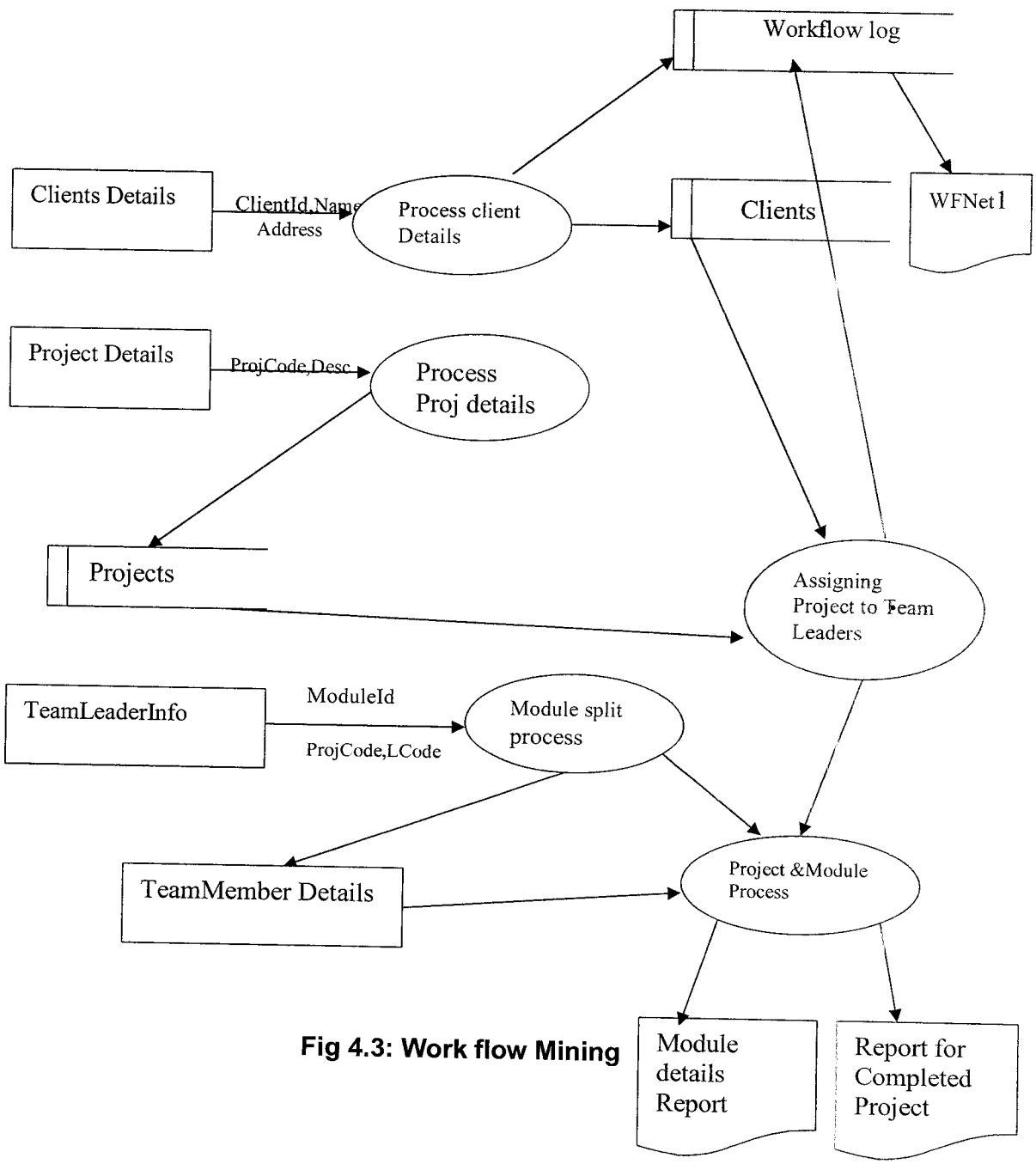


Fig 4.3: Work flow Mining

Implementation

CHAPTER 5

IMPLEMENTATION

Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system

In this project as explained in the System design stage the Project management activity for Software Concern is implemented in VB .NET from this Work flow log is generated as Report. This as input Process Model is constructed with Association Rule algorithm of Data mining to extract knowledge out of WorkFlow Log.

Test cases are performed and its result is matching with expected results.

5.1 SYSTEM VERIFICATION

System Verification answers the question "Am I building the product right?" It includes the review of interim work steps and interim deliverables during a project to ensure they are acceptable. Verification also determines if the system is consistent, adheres to standards, uses reliable techniques and prudent practices, and performs the selected functions in the correct manner. In data access, it verifies whether the right data is being accessed, in terms of the right place and in the right way.

For e.g., the drop downs gather data from the database, so each dropdowns should be verified whether they are bound to the correct database field. It is done during development of the key artifacts.

Verification is a demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle.

In Result Analysis, verification is done during the development itself. Each database bindings are verified after binding to test whether the control is bound to the right data field.

5.2 SYSTEM VALIDATION

Validation answers the question "Am I building the right product?" This checks whether the developer is moving towards the right product, whether the development is moving towards the actual intended product that was agreed upon in the beginning. Validation also determines if the system complies with the requirements and performs functions for which it is intended and meets the organization's goals and user needs. It is traditional and is performed at the end of the project. In data access, it checks whether we are accessing the right data, in terms of data required to satisfy the requirement.

Validation is performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment. It determines the correctness of the final software product by a development project with respect to the user needs and requirements.

Functional validation is done in the Result Analysis System to check whether each of the functions is done correctly as expected in every page. Each control in a Screen is designed to do some function. These functions are checked against the requirements stated for them.

For e.g., clicking 'Save' button should take the corresponding action of saving the details into the database. Clicking the Edit icon should allow one to edit the contents that are being currently displayed. This level of validation can continue to all the controls in the system. This checking is usually done after the system is developed so that all activities that are affected can be checked.

Field level validation is done in Result Analysis to check whether each of the fields either accepts the data as expected and do the client side validation of data entered. For e.g. a field level validation on a text box would check against the type of data entered and follow rules such as length of entry etc.

The data type validation checks are conducted after the form is submitted. It takes place in the Action Form class of the struts framework. If the validation check fails then the processing stops and the control returns back to the original form that was submitted.

The validation is done in a step-by-step process. First the screen is loaded with the controls. When the user moves between controls on the screen, the validation events for the control that lost the focus are fired and appropriate error messages (if any) are displayed. If the user generates a form save request, the entire form is evaluated for any validation controls that are not valid. If even one control is not valid, the form will not be submitted.

5.3 TESTING

Testing is an important stage in the system development life cycle (SDLC). The test case is a set of data that a system will process as normal input. As its philosophy behind testing is to find errors the data are created with the express intent of determining whether the system will process them correctly.

The proper choice of test data is an important as the test itself. If test data as input are not valid or representative of the data to be provided by the user, then the reliability of the output is suspect.

Test data may be artificial. Properly created artificial data should provide all combinations of values and formats and make it possible to test all logic and transaction path subroutines.

The testing done will differ in nature and will have different objectives at each level. The focus of all testing will be to find errors, but different type of errors are looked for at each level.

The levels of testing in this project will be:

- Unit Testing
- Integration Testing
- System Testing

5.3.1 Unit Testing

Unit testing is the lowest level of testing and its function is to test the functionality of basic unit of software in isolation. This is a white box testing where the most detailed investigation of the internal functions of every individual unit is carried out.

The unit test plan describes the features and functionality that is to be tested for each unit. The purpose of unit testing is to find errors, which could be data or logic related errors and also prove that the individual units are robust and fit for purpose they are developed.

The typical tests that will be carried out during unit test include:

- Data validation to check valid and invalid data entered into a text box.
- Field length check to check the maximum length of the field.
- Errors handling to check appropriate front-end validations are being carried out.
- Database validations to check the data entered in the front end are stored into appropriate table in the database.
- Test to ensure that all paths are traversed and branching takes properly.
- Verify operation outside range values.
- Verify operation at normal value range.
- Ensure that all loops are terminated successfully.
- Identify and remove abnormal termination of all loops.

5.3.2 Integration Testing

Integration testing tests the process of integrating the various modules to form the completed system. Integration starts with a set of units each individually tested in isolation and ends when the entire application has been built. Integration testing verifies that the combined units function together correctly. It facilitates in finding problem that occur at interface or communication between the individual parts.

An integration test plan outlines the process and procedure to be followed for integration testing. Integration testing involves the process of testing two or more tested units that have been fully integrated. The integration testing should look for errors in the following.

- The interfaces between the tested units.
- The function that can be performed by the integrated unit.

Programs are invariably related to one another and interact in total system. Each program is tested to see whether it confirms to related program in the system. Each portion of the system is tested against the entire module with both test and live data before the entire system is ready to be implemented.

A bottom-up strategy will be followed for integration testing. This would involve integrating the bottom units with the calling units and test the calling functions. Bottom-up test assures that the lower level modules are tested before testing the higher-level modules, which invoke them. The global variables were traced such that they hold data related to the current module.

5.3.3 System Testing

After integration testing is completed the entire system is tested as a whole. System testing looks for errors in the end-to-end functionality of the system and also for errors in non-functional quality attributes such as performance, volume, and maintainability. The System Testing is also done to prove the robustness of the system and to validate that the system is fit for purposes as per requirement.

System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing or non-testing leads to errors hence it should be rectified.

Another reason for system testing is its utility as a user-oriented vehicle before implementation. The best program is worthless if it does not meet user needs. Unfortunately, the user's demands are often compromised by efforts to facilitate program or design efficiency in terms of processing time or memory utilization.

Following system testing is acceptance testing or running the system with live data by the actual user. No system design is ever-perfect communication problems, programmers negligence are time constraints create error that must be eliminated before the system is ready for user acceptance testing.

Software Testing is the process of confirming the functionality and correctness of software by running it. Software testing is usually performed for one of two reasons:

1) Defect detection
2) Reliability estimation.

White box testing is concerned only with testing the software product; it cannot guarantee that the complete specification has been implemented. Black box testing is concerned only with testing the specification; it cannot guarantee that all parts of the implementation have been tested. Thus black box testing is testing against the specification and will discover ***faults of omission***, indicating that part of the specification has not been fulfilled.

White box testing is testing against the implementation and will discover ***faults of commission***, indicating that part of the implementation is faulty. In order to fully test a software product both black and white box testing are required.

The problem of applying software testing to defect detection is that software can only suggest the presence of flaws, not their absence (unless the testing is exhaustive). The problem of applying software testing to reliability estimation is that the input distribution used for selecting test cases may be flawed.

In both of these cases, the mechanism used to determine whether program output is correct is often impossible to develop. Obviously the benefit of the entire software testing process is highly dependent on many different pieces. If any of these parts is faulty, the entire process is compromised.

Software is now unique unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible.

The key to software testing is trying to find the myriad of failure modes – something that requires exhaustively testing the code on all possible inputs. For most programs, this is computationally infeasible. It is commonplace to attempt to test as many of the syntactic features of the code as possible (within some set of resource constraints) are called *white box* software testing technique.

Techniques that do not consider the code's structure when test cases are selected are called *black box technique.*

Functional testing is a testing process that is black box in nature. It is aimed at examine the overall functionality of the product. It usually includes testing of all the interfaces and should therefore involve the clients in the process.

Final stage of the testing process should be System Testing. This type of test involves examination of the whole computer system, all the software components, all the hard ware components and any interfaces. The whole computer based system is checked not only for validity but also to meet the objectives.

Conclusion

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In this project, we addressed the workflow rediscovery problem. This problem was formulated as follows: "Find a mining algorithm able to rediscover a large class of sound WF-nets on the basis of complete workflow logs." We presented the _ algorithm that is able to rediscover a large and relevant class of workflow processes (SWF-nets).

Through examples, we also showed that the algorithm provides interesting analysis results for workflow processes outside this class. At this point in time, we are improving the mining algorithm such that it is able to rediscover an even larger class of WF-nets. We have tackled the problem of short loops and are now focusing on hidden tasks, duplicate tasks, and advanced routing constructs. However, given the observation that the class of SWF-nets is close to the upper limit of what one can do assuming this notion of completeness, new results will either provide heuristics or require stronger notions of completeness (i.e., more observations). It is important to see the results presented in this paper in the context of a larger effort. The rediscovery problem is not a goal by itself. The overall goal is to be able to analyze any workflow log without any knowledge of the underlying process and in the presence of noise. Therefore, we consider the work presented in this paper as a stepping-stone for good and robust process mining techniques.

We have applied Work Flow mining for Project Management in Software Development Company.

6.2 FUTURE ENHANCEMENT

Most companies already collect and refine massive quantities of data. Data mining techniques can be implemented rapidly on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line. When implemented on high performance client/server or parallel processing.

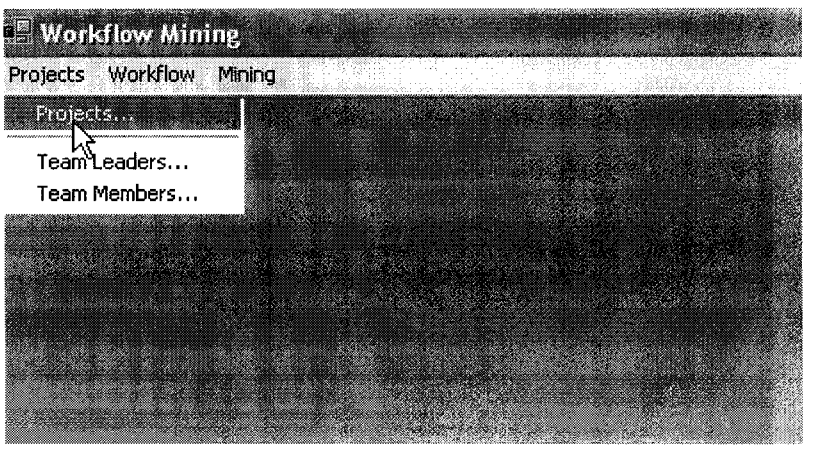
Now a days enormously amount of data are accumulated in all places like departmental stores, shops, hospital, school, college, all type of industries and also in Web. Web contains vast collection of document. It provides answer for almost all the query from variety of user. After the advent of computer the data are enormously available and by making use of such raw collection data to invent the knowledge is the process of Data Mining. Like wise in Web also plenty of Web Documents resides in online. Web is repository of variety of information like Technology, Science, History, Geography, Sports Politics and others. Similar way the routine jobs or work in any industry can be formulated and data store can be produced.

Appendices

APPENDICE

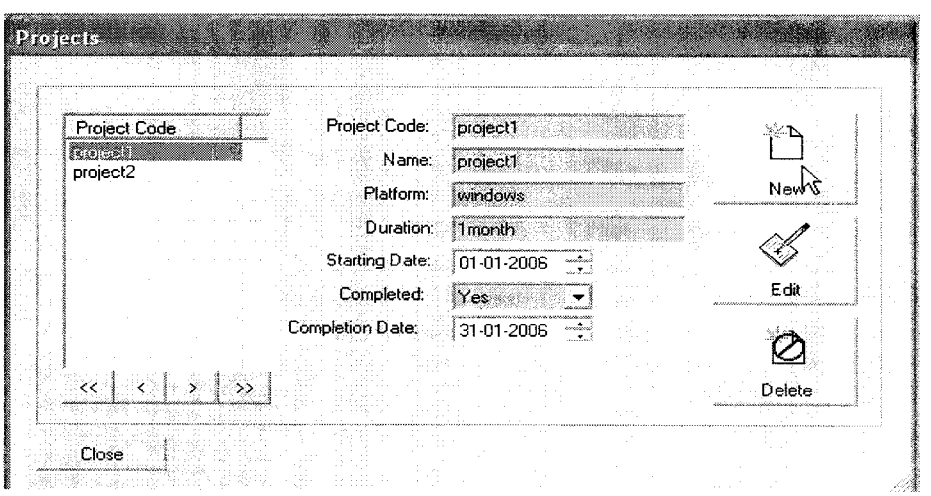
MDI Form – Projects Menu

From Project option, user is given provision to enter Project details, Team leader assigned to project and Team member involved in each project.



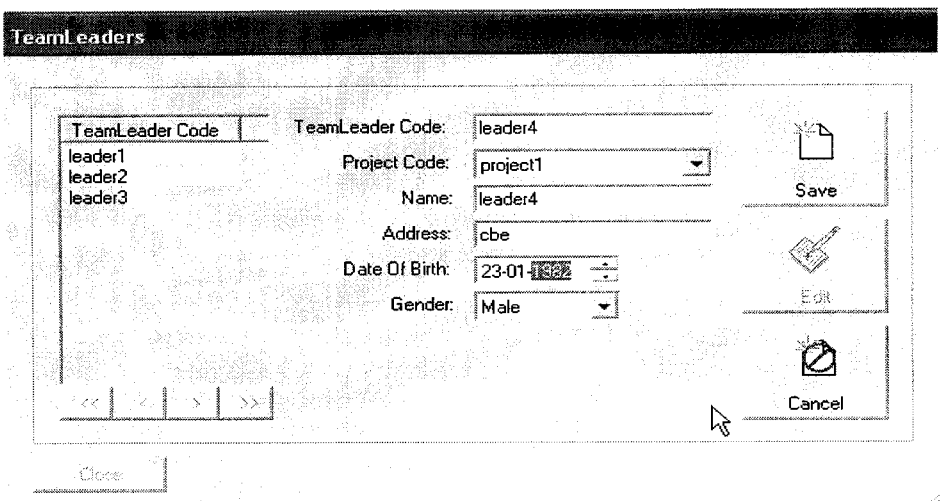
Project Details Form

The details of the project can be stored, edited or deleted as follows:



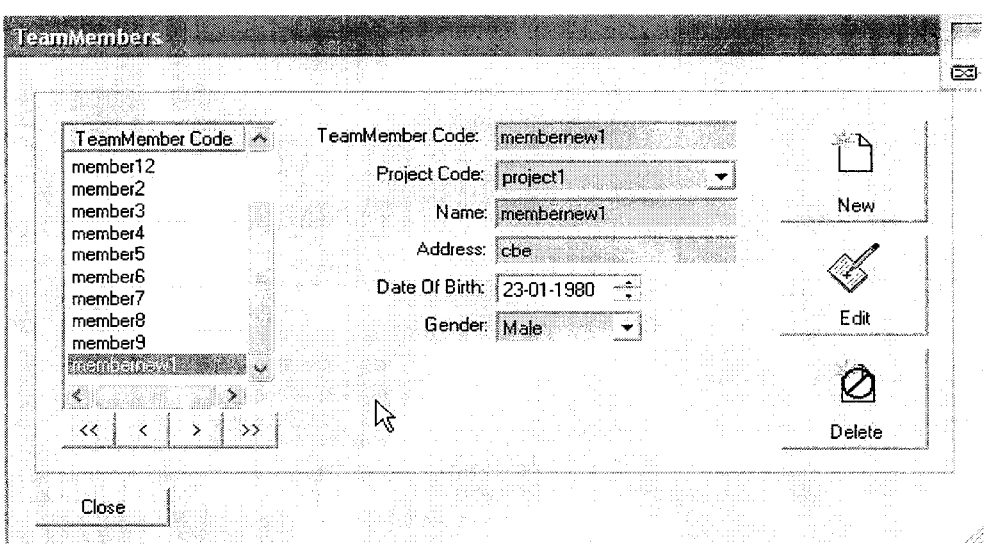
Team Leaders Form

After project is accepted by the company the Team Leader assigned modules as follows:



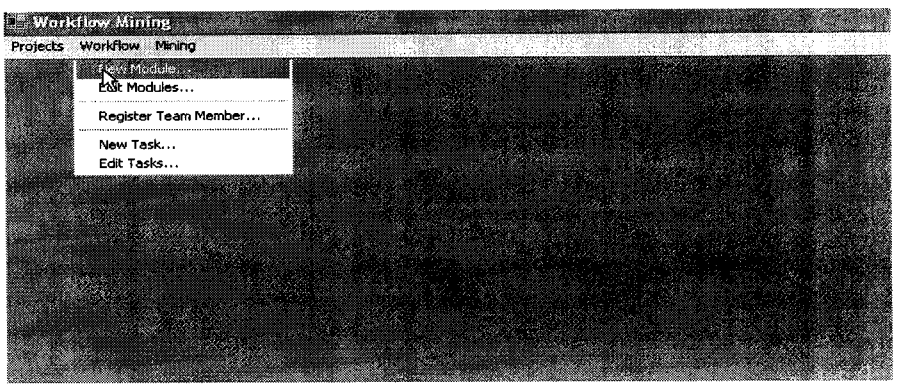
Team Members Form

For each Team Leader, the Team members are assigned as follows:



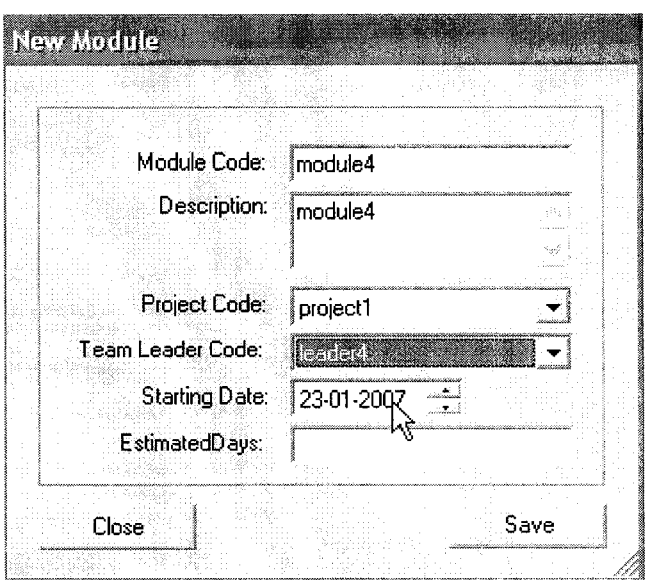
MDI Form – WorkFlow Menu

From the WorkFlow menu option, new module details can be entered and edited, team members can be assigned for each module, and new tasks can be entered for the modules.



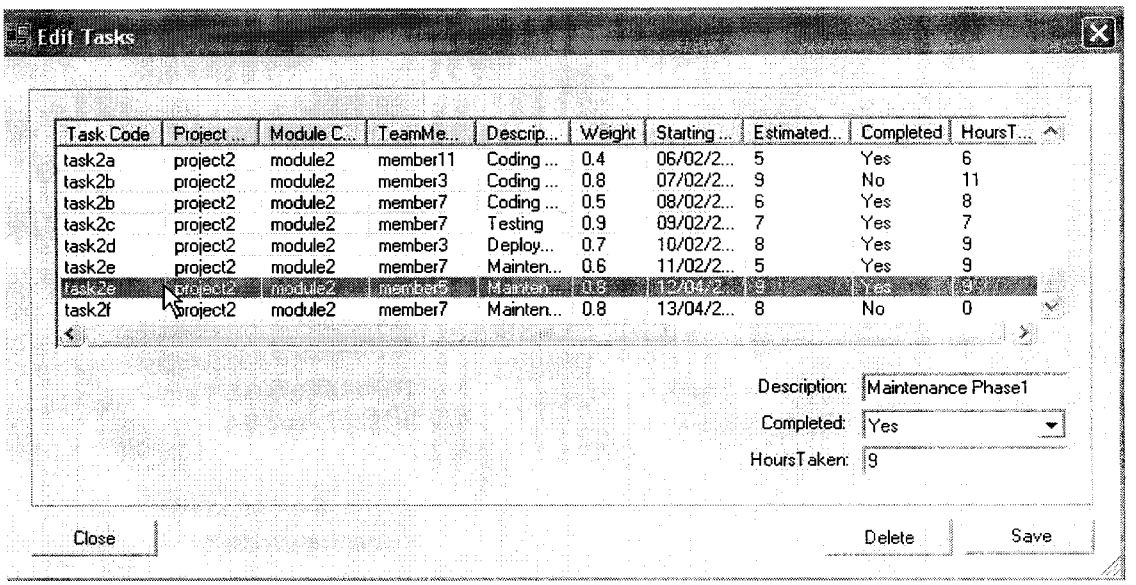
New Module Form

Details of the new module of the project can be entered as follows:



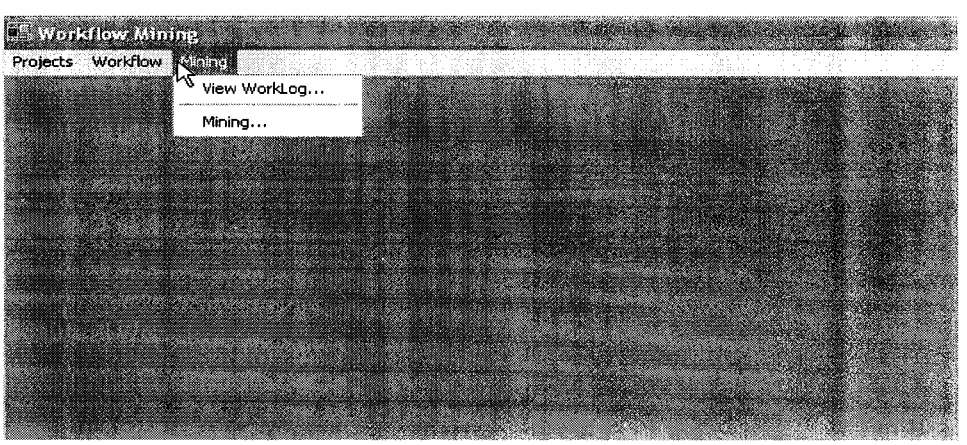
Edit Tasks Form

The details of the tasks can be edited as follows:



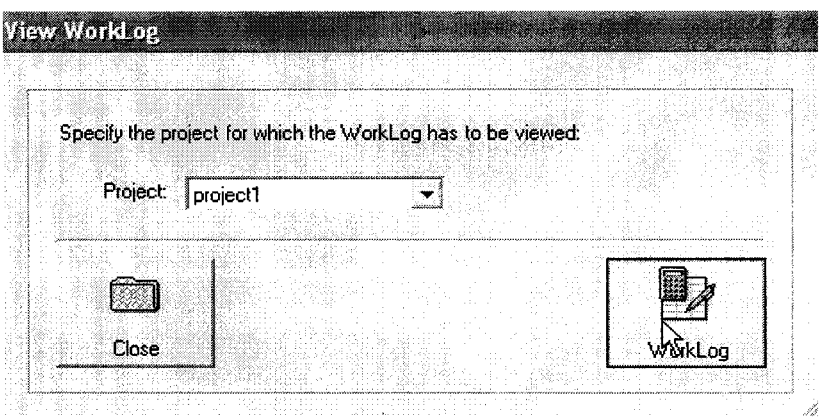
MDI Form – Mining Menu

From the Mining menu option, the WorLog can be generated and viewed using the Project Management data.

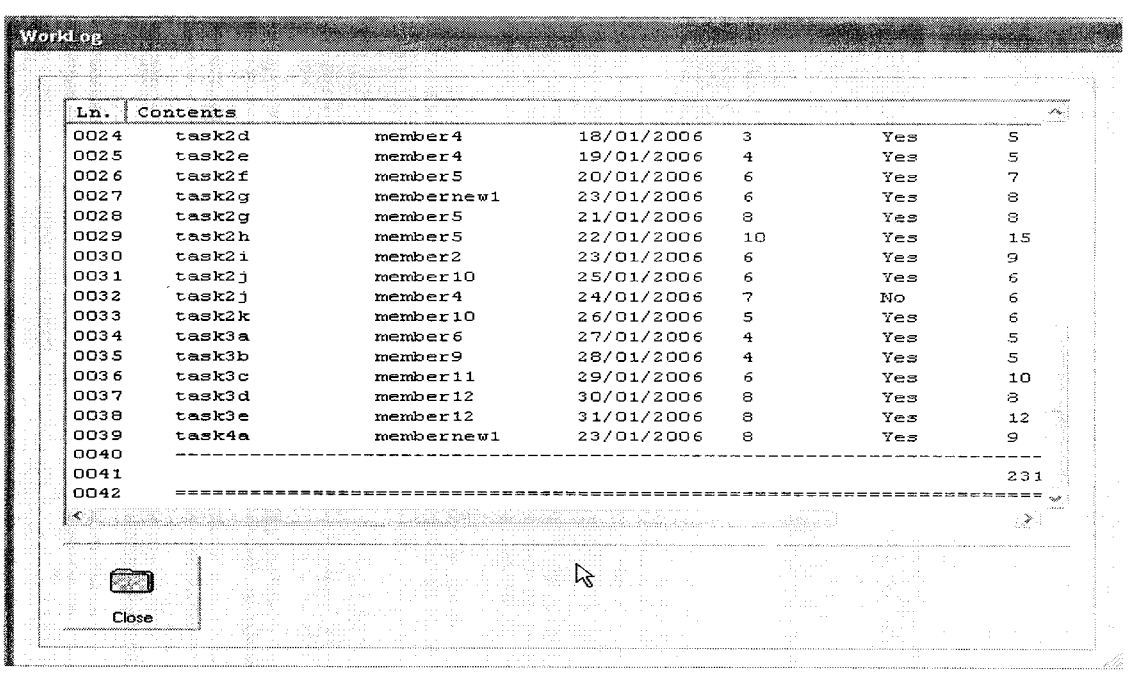


WorkLog Form

The log file of a particular project can be viewed by giving the project name as input.

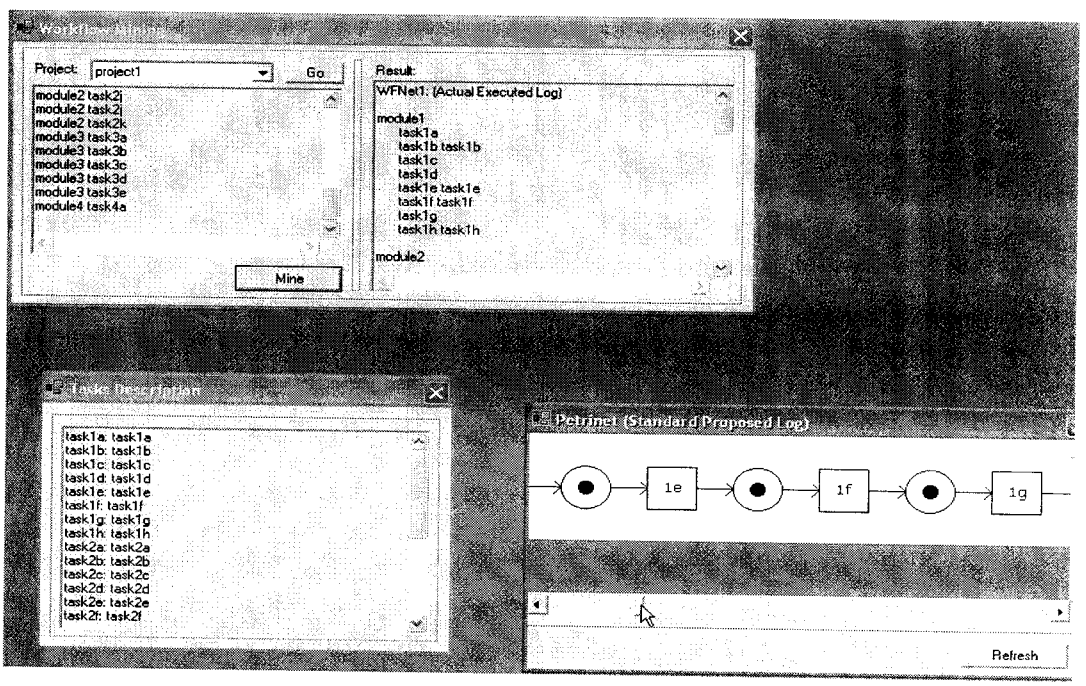


The log file of the project specified is given as follows:



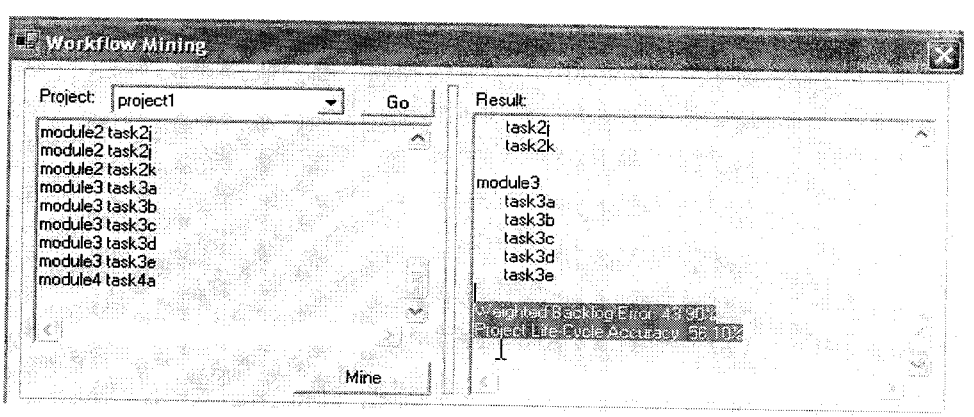
Petrinet – Standard Proposed Log

With the modules and tasks information of the project, mining process is performed and a Workflow net called Standard Proposed Log is constructed.



Mining Form

The Weightage of the BackLog Error and the Project Life Cycle Accuracy is displayed by comparing the two WorkFlow nets.



Reference

REFERENCE

1. Elias M. Awad (2000) "Systems Analysis and Design", Galgotia Publications.

2. Matthew MacDonald (2002) "The Complete Reference VB.NET ", Tata McGraw-Hill Publishing Company Limited.

3. Roger S. Pressman (2000) "Software Engineering ", McGraw-Hill Publishing Company Limited.

4. Steven Holzner (2002) "Visual Basic .Net Programming " Black Book -Dream Tech Press, New Delhi.