

**AN INTEGRATED FRAMEWORK FOR
OFFICE AUTOMATION**

By

YE-QIONG ZHU

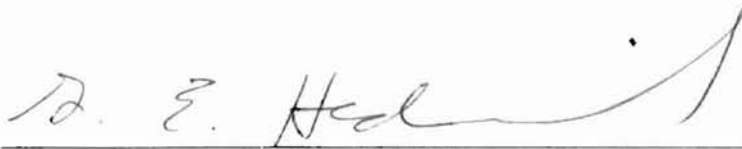
Bachelor of Engineering
Shanghai Technology Institute
Shanghai, P. R. China
1983

Master of Business Administration
Oklahoma State University
Stillwater, Oklahoma
1993

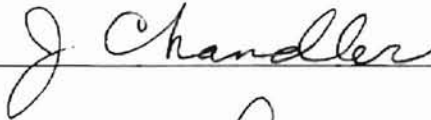
Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
In partial fulfillment of
The requirements for
The Degree of
MASTER OF SCIENCE
July, 1999

AN INTEGRATED FRAMEWORK FOR
OFFICE AUTOMATION

Thesis Approved:



Thesis Advisor







Dean of the Graduate College

ACKNOWLEDGEMENTS

I sincerely thank my principal advisor Dr. G. E. Hedrick, for his guidance, help, encouragement, and continuous support in finishing this thesis. My deepest appreciation is also addressed to my other committee members Dr. John P. Chandler and Dr. Huizhu Lu for their advice, cooperation, and insightful suggestions during the development of this project.

So many people have helped me throughout the process of idea generation to the completion of this thesis, it is impossible to acknowledge them all personally. I extend my special thanks to Dr. Robert Brown, Director, Arts & Sciences Extension for his unceasing support during all the phases of this project. Dr. Lu and her team of research associates/ programmers, especially Hung-Chi Su, have contributed very professionally to the planning as well as execution of this project as an interdepartmental contractor of Oklahoma State University. My respectful thanks for Dr. Manjunath Kamath and Partha Ramachandran, Department of Industrial Engineering and Management for helping me understand the intricacies of Petri Net from an engineering perspective.

My special gratitude is extended to my husband Dr. Masood Rahman, and my children Effie and Sophia for their patience, support, and encouragement during my studies. I would also like to express my appreciation to all those people who have contributed by giving many valuable suggestions.

Finally, I would like to dedicate this thesis to my mother Dr. Wang and late father Dr. Enliang Zhu who have continuously supported my enthusiasm for higher education.

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
Motivation	3
Problem Statement and Intent	3
Organization of the Thesis	4
II. LITERATURE REVIEW	6
Office Automation Defined	6
The Evolution of Office Automation Research	7
Local Computing	8
Workgroup Computing	9
Workflow Computing	9
Intelligent Workflow Computing	13
Office Automation Modeling and Analysis Methods	14
Process Modeling Using Petri Net	14
The High-Level Petri Nets	17
The Properties of Petri Net	19
Other Process and Data Modeling Methods	20
System Modeling Method	20
Data Modeling Method	22
Organizational Aspects of Office Automation	23
III. A CASE STUDY – AUTOMATING THE MANAGERIAL AND FUNCTIONAL OPERATIONS OF A&S EXTENSION OFFICE, OSU	24
Extension Office Characteristics	24
Procedure Analysis of Basic Functional Units	25
Pre-Integrated Automation	32
The Decision to Automate and Process Reengineering	34
Design Objectives	37

Workflow Redesign Using Petri Nets	38
General Model of Extension Office	38
Marketing Decomposition	43
Planning and Budgeting Decomposition	44
Registration Decomposition	45
Program Preparation and Billing Decompositions	46
Financial Accounting Decomposition	47
Managerial Decision Making Decomposition	48
The Architecture of Automation	49
Existing Software Tools	49
FoxPro Database	50
PCs and MACs	50
External Databases	51
IV. WORKFLOW VALIDATION AND VERIFICATION	52
Validation via Simulation	53
Verification via Net Property Analysis	56
Reachability	57
Boundedness	60
Liveness	60
Reversibility	61
Persistence	61
Conservative	62
V. OTHER ISSUES OF OFFICE AUTOMATION	66
Attitude and Expectations	66
User-Driven Approach	67
Office Automation and Productivity	69
Office Automation as Hindrance	69
VI. SUMMARY AND FUTURE WORK	71
Summary	71
Further Research Topics	72
BIBLIOGRAPHY	73
APPENDIXES	79
APPENDIX A – THE EXTENSION WORKFLOW NET IN TEXT FORM OF VIEW	80

APPENDIX B – THE REACHABILITY TREE IN GRAPHICAL VIEW	81
APPENDIX C – THE REACHABILITY TREE IN TEXT FORM OF VIEW	96
APPENDIX D – ALL MARKINGS OF THE SYSTEM	98

LIST OF TABLES

Table		Page
I.	The Flow Matrix of the Extension Workflow Net	54
II.	The Change Matrix of the Extension Workflow Net	55
III.	The T-Invariant of the Extension Workflow Net	64
IV.	The P-Invariant of the Extension Workflow Net	64
V.	Partial Markings of the Extension Workflow Net	65

LIST OF FIGURES

Figure	Page
1. An Example Petri Net	17
2. Hierarchical Decomposition	18
3. The Basic Functional Unit of A&S Extension Office	26
4. Extension Office Workflow Diagram	34
5. Petri Net Model for Extension Office	39
6. Sub-Net of Marketing	43
7. Sub-Net of Budgeting	44
8. Sub-Net of Enrollment/Payment	45
9. Sub-Net of Program Preparation	46
10. Sub-Net of Billing	46
11. Sub-Net of Financial Accounting	47
12. Sub-Net of Workload Assessment	48
13. Architecture of A&S Ext. Office automation	49
14. Net Properties Calculated by the Reachability Tree of the Workflow	58
15. Net Properties Calculated by the Invariants of the Workflow	58
16. Net Properties Calculated by the Marked Graph of the Workflow	59
17. Net Properties Calculated by the Free-Choice Net of the Workflow	59
18. The Reachability Tree of the Extension Workflow Net (Partial)	63

NOMENCLATURE

PN	Petri net
P	$\{ p_1, \dots, p_n \}$, a finite set of places
T	$\{ t_1, \dots, t_m \}$, a finite set of transitions
I	input mapping $P \times T \equiv \{ 0, 1 \}$
O	output mapping $P \times T \equiv \{ 0, 1 \}$
X	$P \cup T$
p_i	places in Petri net
t_i	transitions in Petri net
$\cdot t$	the set of all input places to transition t
$t \cdot$	the set of all output places from transition t
$\cdot p$	the set of all input transitions to place p
$p \cdot$	the set of all output transitions from place p
$m(p)$ or m	The marking of a Petri net
$R(m_0)$	The set of all possible markings reachable from m_0
W	A workflow
tw_i	tasks in workflow
OP_i	operations in workflow
(b_i)	task primitive: begin
(a_i)	task primitive: abort
(p_i)	task primitive: pre-commit

(c _i)	task primitive: commit
(in _i)	initial state
(ex _i)	execution state
(dn _i)	done state
(cm _i)	commit state
(ab _i)	abort state
W ^s	a workflow state

CHAPTER I

INTRODUCTION

Office Automation has had a tremendous impact on both business and public organizations over the past twenty years. Besides supporting different levels of office tasks, office automation systems are becoming an effective medium for today's distributed work environment (Ellis and Gibbs, 1991)

A number of office automation analyses and design methodologies have appeared in the literature, each of them has its strengths and weaknesses. Most existing methodologies have difficulty describing and analyzing the complexity of office work (Bracchi and Pernici 1984, Hewitt 1986). Various techniques of structured analysis, including those based on an object-oriented paradigm, focus principally on the analysis and design of structured and semi-structured activities, and are used essentially for systems specification (Martin and McClure 1988, Nierstrasz 1992, Scheer 1992). At the other extreme, the Critical Task Methods (Harris and Brightman 1985) emphasizes unstructured or ill-structured work. Office Analysis and Diagnosis Methodology (Sirbu and Schoichet 1983, Sutherland 1983) is targeted at semi-structured office activities.

Automation technology is not new to organizations. As early as the 1920's the automatic telephone switchboards were in use (Epstein 1991). But widespread growth of automation did not become practical until after World War II when some major

theoretical breakthroughs were made and the electronic computer was developed. By the mid-1950's major industries were using electronic computers in their operations. For instance, the oil industry was using automation to control oil tank farms and the commercial industry was using the computer for payroll and loan processing (Law and Gorla 1996).

Today office automation has spread to the corporate world and non-profit organizations as the advances in computer and telecommunications technology have led to the development of "office automation," also referred to as "office information systems." However, most of the early developments in office automation were aimed at increasing the productivity of the frontline office workers such as secretarial level staff. It did not address the need for an integrated system in which a diverse group of office professionals ranging from hourly paid receptionists to activity planners to mid-level management who could interact simultaneously in the decision making processes of an office environment. Each level operated on its own as an isolated and local system. There is a lack of information exchange and significant gaps in the stream of workflow among various elements of an office. As we look at the office as a process, there is clearly the need for a coherent, integrated office automation system.

There are means to handle the complexity of office environment, especially the kinds of activities and processes that take place in today's organization. One that is receiving increasing attention is through the use of workflow (Peterson 1981) and Petri nets (Reisig 1986). The objective of this thesis is to describe an integrated framework of office automation using Petri net based workflow technology. The analysis, simulation, and validation of the framework under study are presented and discussed. An actual case

study of the implementation of an office automation system is presented. The issues of workflow redesign and modeling using Petri net, as well as the issues related to the software engineering/development in organizational environment are discussed. Limitations of the framework and future research directions are mentioned also.

Motivation

The motivation for this thesis is to explore the design, analysis, and implementation of an integrated office automation system using Petri net based workflow technology. The automation system could replace the current manual operation system employed in the organizations with the objective of being flexible, expendable, and user friendly.

Problem Statement and Intent

An extension office in a major state university decided to implement office automation in a non-technical staff environment to increase staff productivity. While the staff had been using office automation tools such as Microsoft Word, Microsoft Excel, Power Point, PageMaker, etc., there was lack of a coherent and integrated system. Each staff member, ranging from the receptionist to the Director, operated on his/her own PC or MAC, and created his/her own data files. The result is a classical case of isolated operational islands. There was a lack of information exchange and significant gaps in the stream of workflow among various functional units of the office. Data redundancy,

inconsistency, and repetition were inevitable. For example, the same registration information would be keyed in at least three times during registration period, tuition payment period, and accounting entries period. Due to data inconsistency, confusion arose. For instance, a student had dropped a course in the middle of a semester, while office records showed that he/she was still enrolled. Instructors' pays were delayed and deadlines were missed due to lack of timely information. The program coordinators planned their own program schedules and allocated physical resources independently, which resulted into a variety of bottlenecks leading to conflict of timings, resource allocations, and occasional slowing of office operations.

In spring 1997, the registration and accounting system of the Extension office crashed. It was a stand alone 1986 DOS-driven software package developed by a small consulting company. The package had been upgraded twice during the ten years period to bring it up to a network environment. In early 1994 the company claimed that it would no longer supporting the package. The Extension office thus had to decide whether to repair the software or to switch to an up-to-date office automation system.

Organization of the Thesis

The thesis is organized into six chapters. Chapter I is an introduction on the background and the motivation of office automation as the thesis topic. An actual case problem is presented for discussion.

Chapter II reviews the literatures on the topics of office automation definition, the evolution of the research on office automation, and the developments on various office

modeling and analysis techniques. The definition of Petri Net and its basic properties are given. The literature on the impact of automation in office environment is also mentioned so as to view office automation from a broader perspective.

Chapter III discusses an actual case study. The outline of the case and the automation solution to the problem is presented. Petri nets as a modeling tool for office workflow redesign is discussed in detail. The hierarchy feature of the Petri net is used to develop various levels of details for the system. The architecture of the automation system is presented as well.

Chapter V discusses the workflow simulation, analysis, and verification of the Petri net based workflow system. The simulation, analysis, and verification of an automation system before its actual development can identify possible problems such as deadlock in the system, ensuring that the system proposed is correct and reliable.

Chapter IV outlines the key factors that could decide the success or failure of office automation systems based on the case study in chapter III. In many situations, office automation systems were developed successfully from technical point of view, but it failed in organizational application due to the key factors mentioned here.

Chapter VI gives a summary of the thesis as well as further research topics in the area of office automation.

CHAPTER II

LITERATURE REVIEW

Office Automation Defined

The primary effort of Office Automation in 1970's was on increased productivity of clerical and secretarial workers. Recent attention has been aimed at how both management and professionals can benefit from office automation through better information during the process of decision-making, and yield an overall productivity increase. (Goldfield 1985, Gremillion 1985).

Jarrett, D. (1984) defined Office Automation as the application of electronic and electro-mechanical devices for the purpose of increasing the productivity of office workers. The devices include general-purpose computers, special-purpose office machines and communications equipment. The office workers include managers, professionals, secretaries and clerks. The term "knowledge workers" has been used to describe both managers and a variety of professionals. Secretarial/clerical employees achieve increased productivity in their ability to create, store, retrieve and transmit business communications. Knowledge workers achieve increased productivity in terms of information and quality decisions (McLeod and Jones, 1987).

As the design of office automation systems relies upon underlying conceptions of human organization and action, and its representation of office work and objects, Hewitt (1986) defined the nature of office work from an open systems perspective. Coping with conflicting, inconsistent, and partial information is one of the major challenges in office automation systems. Gerson and Star (1986) proposed due process as a central activity in organizational information processing. A due process is the process of articulating alternative solutions. There is no formal solution to the problem of due process as a result of multiple competing, possibly irreconcilable, solutions.

Hirschheim (1986) argued that office automation could not be successfully applied without a detailed understanding of the environment into which it is placed. The goal of office automation requires a representation of office work and its objectives. Suchman (1983) pointed out that the procedural structure of an organizational activity is the product of the orderly work of the office, rather than the reflection of some enduring structure that stands behind that work. It is structured social process in the sense that it is the process produced by participants at particular times and places.

The Evolution of Office Automation Research

Office Automation research has developed considerably over the years from local computing systems to intelligent workflow in a network of computing systems. By reviewing the evolution of Office Automation research, we can better understand the current efforts in Office Automation system research and show how we can benefit from the earlier work in this area.

Local computing and group computing primarily have been oriented toward office work that is structured and organized (Croft and Lefkowitz, 1984). With the difficulties encountered in modeling unstructured office work, workflow computing and intelligent workflow computing offer the most promise of delivering effective support for the office automation system. Workflow computing opened a way toward an unstructured work profile that best matches the actual office situation as an open system. As Mahling and Craven (1995) pointed out, most endeavors in business or daily life can be described as processes. These processes consist of tasks that usually involve more than a single person or intelligent machine agent. Office automation could not be successfully applied without a detailed understanding of the environment into which it is placed (Hirschheim 1986).

Local Computing

The origin of office automation system based task-support for individuals and groups dated back to the early days of computing machinery. The limited business success of the “paperless office” and other process based systems in the 1970s and 1980s lessened interest in this area. Local computing automated routine activities and assisted in complex tasks in an office. Its goal was to move the paperwork through the office (Ang, Vanneste, and Lu 1998). There was no sharing of information between different departments of an organization and was procedurally rigid, resulting in limited applicability.

Workgroup Computing

Activity in groupware research in the 1980s and 1990s shifted to the definition aspects of cooperative work, such as shared information spaces or shared editing. One of the major problems in interactive process support is spatial and temporal distribution (Casati, Ceri, and Pozzi, 1998). The competence of the cooperating partners may vary widely. Viewing human and machine agents as partners in communication, cooperation, problem solving and task execution extends the conventional paradigm of computer supported cooperative work and distributed Artificial Intelligence (Gruhn 1995).

Workgroup Computing facilitates the transfer and sharing of information from workgroup to workgroup or individual to individual. It is a “information centered” model that focuses on the information being processed instead of process (Koulopoulos 1995). Compared with local computing, workgroup computing focuses on enhancing the user’s ability to share information within workgroups.

Workflow Computing

In the 1990’s the workflow concept started to emerge as a system approach to office automation technology. In contrast with workgroup computing, workflow computing emphasizes the importance of the process, which acts as a container for the information (Rusinkiewicz 1998). In this way workflow combines rules, which govern the tasks performed, and it coordinates the transfer of the information required to support these tasks. It is a “process-centered” model.

Workflow computing applies many of the same concepts and benefits of factory Automation and industrial engineering to the process of work management in an office Environment. Tomas Koulopoulos (1995) defined workflow as “a tool set for the proactive analysis, compression, and automation of information-based tasks and activities.” The basic premise of workflow is that an office environment is an information factory, or more specifically, a process factory (Sheth and Georgakopoulos, 1996). The process, which can exist in a range of formats from paper to electronic form, provides the basic raw material of every office task. The connection of these office tasks creates a value chain – a series of tasks or processes which add value to an information object as it progresses from task to task – that spans internal and external task boundaries. In this architecture, workflow attempts to streamline the components of the document factory by eliminating unnecessary tasks, thereby saving time, effort, and costs associated with the performance of those tasks and automating the remaining tasks that are necessary to a process.

Numerous applications that use workflow computing systems in their day-to-day activities for controlling their business processes exist today. These include office automation, healthcare, telecommunications, etc. A workflow is a representation of a given process that is comprises of well-defined collection of activities, referred to as tasks (Georgakopoulos, Hornick and Manola, 1996). Each of these tasks, serving a given function in the overall process, has certain information input requirements and produces information as part of its output.

Adam, Atlure, and Huang (1998) gave following formal definition of workflow.

A workflow, W , can be represented as a set of tasks tw_1, tw_2, \dots, tw_n , together with their order of invocation and information flow (Georgakopoulos and Hornick, 1995). A task t_i consists of a set of operations OP_i and task primitives $\{ \text{begin } (b_i), \text{ abort } (a_i), \text{ precommit } (p_i), \text{ commit } (c_i) \}$. Execution of a task, in addition to invoking operations that may require access to data items, requires invocation of the task primitives. All operations in a task must be executed only after the begin primitive is issued. All operations must end with either a commit or abort. At any given time, a task can be in one of the following states: initial state (in_i), execution state (ex_i), done state (dn_i), commit state (cm_i), or abort state (ab_i). A primitive may move a task from one state to another. For instance, a task t_i may move from its initial state in_i to the execution state ex_i by invoking the begin primitive b_i . The commit primitive c_i should follow the precommit primitive p_i . This particular representation of task with the precommit primitive is required to enforce some types of dependencies. Formally, a task can be defined as follows.

Definition 1. A task t_i is a partial order with an ordering relation \prec_i where

1. $t_i = OP_i \cup OP_i$ where $OP_i \subset \{ b_i, p_i, a_i, c_i \}$;
2. $c_i \in t_i$ iff $b_i \in t_i \wedge a_i \notin t_i$, and $a_i \in t_i$ iff $b_i \in t_i \wedge c_i \notin t_i$; and
3. for any $op_i \in OP_i$, $b_i \prec_i op_i \prec_i$ either p_i (if $c_i \in t_i$) or a_i (if $a_i \in t_i$).

As a consequence of the above definition, we can deduce the following:

1. For any task t_i , one of the following precedence relationships holds:

$$in_i \prec ex_i \prec dn_i \prec cm_i, in_i \prec ex_i \prec dn_i \prec ab_i, \text{ or } in_i \prec ex_i \prec ab_i.$$

2. There is no precedence order imposed between the states cm_i and ab_i and in fact are two mutually exclusive states. We say cm_i is a complementary state of ab_i , and vice versa (denoted as $cm_i = ab'_i$ and $ab_i = cm'_i$). Thus we define an incompatible state set for a task = $\{ cm_i, ab_i \}$, which states that tasks in this set cannot coexist at any given time.

Definition 2. Given a workflow $W = \{ tw_1, tw_2, \dots tw_n \}$ we define a workflow state W^s as the set of current states of each tw_i in W .

For instance, suppose a workflow consists of three tasks tw_1 , tw_2 , and tw_3 , a workflow state W^s could be $\{ ex_1, cm_2, ab_3 \}$, meaning that tw_1 is in execution state tw_2 in commit state and tw_3 in abort state.

Intelligent Workflow Computing

As office automation is moving from purely structured systems to semi-structured systems, workflow computing has opened a way for the “open” or unstructured systems that characterize the office processes (Shan, Davis, and Du, 1996). Various intelligent workflow systems appeared in literature as well as applications:

Agent-Enhanced Workflow employs a community of intelligent, distributed and autonomous software agents to improve the management of business processes. It allows the software agents to negotiate with each other to establish contracts that govern the distribution of work across the processing centers (Judge, Odgers, Shepherdson, and Cui 1998)

Knowledge-based workflow goes beyond the simple rule-based approach to provide methods of incorporating exception processing. In a knowledge-based workflow system the workflow would use statistical, heuristic, and artificial intelligence to infer correct routing, scheduling, and exception routing (Obrien and Wiegand, 1996).

Object-Oriented Workflow (Chen and Lu 1997, Reichert and Dadam, 1998) combines workflow technology with object-oriented technology. It signifies what may be a fifth generation of applications development tools based on an object-oriented workflow metaphor (Hus and Kleissner, 1996).

Office Automation Modeling and Analysis Methods

Process Modeling Using Petri Nets

The classical Petri net was invented by Carl Adam Petri in the sixties. Since then Petri nets have been used to model and analyze all kinds of processes with applications ranging from protocols, hardware, and embedded systems to flexible manufacturing systems, user interaction, and business processes (Voss 1986). In the last two decades the classical Petri net has been extended with color, time and hierarchy. (Jensen 1996, Van Der Aalst 1994). These extensions facilitate the modeling of complex processes where data and time are important factors. Van Der Aalst (1998) summarized the reasons for using Petri nets for workflow:

- **Formal Semantics**
A workflow process specified in terms of a Petri net has a clear and precise definition, because the semantics of the classical Petri net and several enhancements (color, time, hierarchy) have been defined formally.
- **Graphical Nature**
Petri nets are a graphical language. As a result, Petri nets are intuitive and easy to learn. The graphical nature also supports the communication with end-users.
- **Expressiveness**
Petri nets support all the primitives needed to model a workflow process. Almost all the routing constructs present in today's workflow management system can be modeled.
- **Properties**
In the last three decades many people have investigated the basic properties of Petri nets. The firm mathematical foundation allows for the reasoning about these properties. There is a lot of common knowledge about this modeling technique.
- **Analysis**
Petri nets are marked by the availability of many analytical techniques. This is a great asset in favor of Petri nets for workflow modeling. These techniques can be used to

prove properties (safety, invariance, deadlock, etc.) and to calculate performance measures (response times, waiting times, occupation rates, etc.).

An office workflow system, in its general form, is a heterogeneous and distributed information system where the tasks are performed using an autonomous system. Resources, such as databases, labor, etc. are typically required to process these tasks (Atlure and Huang, 1996). By decomposing each task into a number of primitives and representing the states in between these primitives, we can model a workflow as a general Petri net; thereby, enable us to adopt the already available analytical tools and techniques for analyzing workflows (Ellis and Nutt, 1993).

A formal definition of Petri Net given by Desrochers and Al-Jaar (1995):

An ordinary Petri net is a bipartite directed graph represented by a quadruple $PN = (P, T, I, O)$, where $P = \{p_1, \dots, p_n\}$ is a finite set of places, $T = \{t_1, \dots, t_m\}$ is a finite set of transitions, I is an input mapping $P \times T \equiv \{0, 1\}$ corresponding to the set of directed arcs from P to T , and O is an output mapping $T \times P \equiv \{0, 1\}$ corresponding to the set of directed arcs from T to P .

A *node* represents either a place or a transition in a Petri net. $X = P \cup T$ represents the set of nodes in the Petri net

Define $\cdot t = \{ \forall p \in P \mid I(p, t) = 1 \},$

the set of all input places to transition t , and

$$t \cdot = \{ \forall p \in P \mid O(t, p) = 1 \},$$

the set of all output places from transition t . $\cdot t$ will be called the *preset* of t , and $t \cdot$ will be the *postset* of t . Similarly, let

$$\cdot p = \{ \forall t \in T \mid O(t, p) = 1 \},$$

the set of all input transitions to place p , and

$$p^{\bullet} = \{ \forall t \in T \mid I(p, t) = 1 \},$$

the set of all output transitions from place p . These are referred to as the preset and postset of p .

A *pure Petri net* is a net such that $\forall t \in T, \{ \cdot t \} \cap \{ t \cdot \} = \emptyset$; i.e., no self-loops. Any Petri net which is not pure can be transformed into a pure Petri net by adding appropriate dummy places and transitions.

Tokens (represented graphically by dots) are assigned to places and can be used to represent several different modeling situations. For example, if a place represents a resource, then the existence of a token in that place means that the resource is available. Tokens are also used to represent a logical condition. A token in a place can be used to imply that the condition is true, and no token, that it is false.

The *marking* $m(p)$ or m of a Petri net is a mapping, $m:P \rightarrow \{0, 1, 2, \dots\}$, that assigns a finite nonnegative integer number of tokens to each place.

Assuming that the weight of each arc is 1, a transition t is said to be enabled with respect to a given marking m , if $m(p_i) > 0, \forall p_i \in \cdot t$. An enabled transition can fire by removing one token from each of its input places, and adding one token to each of its output places. The firing of a transition results in a new marking, m_1 , where

$$m_1(p_i) = \begin{cases} m_0(p_i) - 1 & \text{if } p_i \in \cdot t \\ m_0(p_i) + 1 & \text{if } p_i \in t \cdot \\ m_0(p_i) & \text{otherwise} \end{cases}$$

The firing of a transition t is denoted by: $m_0 \xrightarrow{t} m_1$

Assuming transition t_j fires, $j = 1, 2, \dots, m$, then above expression can be written as

$$m_1(p_i) = m_0(p_i) + O(t_j, p_i) - I(p_i, t_j) \quad \text{for all places } p_i, I = 1, 2, \dots, n.$$

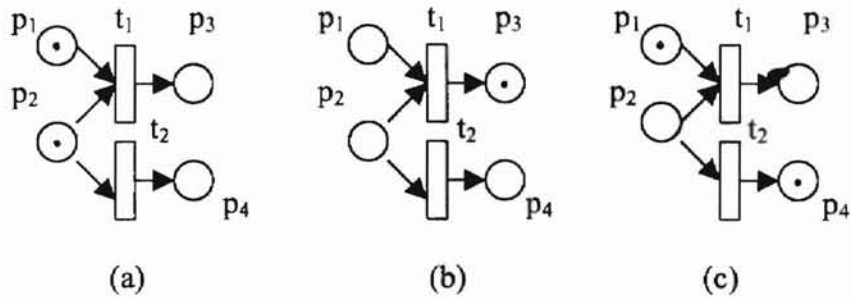


Figure 1: An example Prtri net: (a) initial marking (b) if t_1 fires first (c) if t_2 fires first

High-level Petri Nets

The classical Petri net allows for the modeling of states, events, conditions, synchronization, parallelism, choice, and iteration. However, Petri nets describing real world processes tend to be complex and extremely large. Moreover, the classical Petri net does not allow for the modeling of data and time. To solve these problems, many extensions have been proposed. A Petri net extended with color, time, and hierarchy is called a high-level Petri net.

The extension with color to model data (Jensen 1996). In colored Petri net tokens often represent objects (resources, goods, humans), and color the attributes of these objects or the value of the tokens. Transitions determine the values of the produced tokens on the basis of the values of the consumed tokens. It is also possible to specify “preconditions” which take the colors of tokens to be consumed into account.

The extension with time (Van Der Aalst 1998). In real systems it is often important to describe the temporal behavior of the system, Since the classical Petri net is not capable of handling quantitative time, a timing concept is added. There are many

ways to introduce time into Petri net, such as associating time with tokens, places, and/or transitions.

The extension with hierarchy to structure large models (Van Der Aalst 1998). Although timed colored Petri nets allow for a succinct description of many business processes, precise specifications for real systems have a tendency to become large and complex. Thus a hierarchy construct – subnet is added. A subnet is an aggregation of a number of places, transitions, and subsystems. It can be used to structure large processes. At one level we want to give a simple description of the process (without having to consider all the details). At another level we want to specify a more detailed behavior. The extension with hierarchy allows for such an approach.

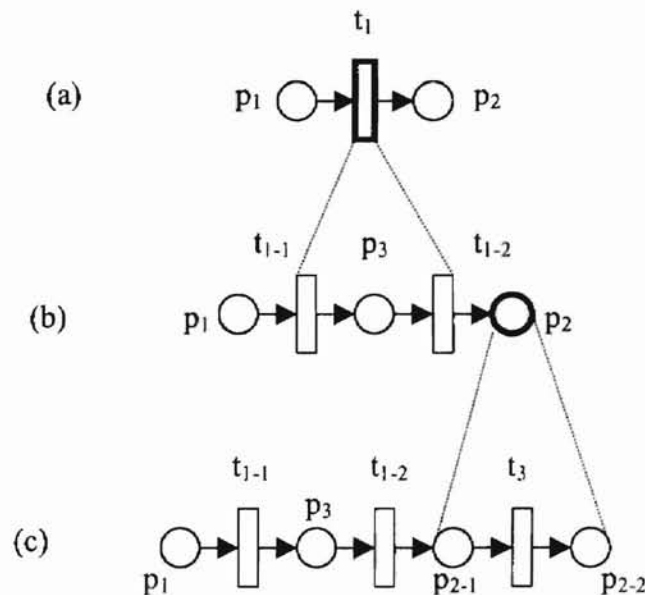


Figure 2. Hierarchical decomposition: (a) system description at aggregated level (b) system description after decomposing t_1 (c) system description after decomposing p_2 .

The Properties of Petri Net

Dicesare and Harhalakis (1993) gave following definitions of basic behavioral properties of Petri net:

- Reachability - The marking m_r is said to be reachable from m_0 if there exists a firing sequence that will yield m_r . The set of all possible markings reachable from m_0 is called the reachability set and is denoted by $R(m_0)$. Given a Petri net, it is necessary to know if there is a sequential firing of transitions that can bring the net to a new marking m_r .
- Boundedness - A Petri net is k -bounded with respect to an initial marking m_0 if each place in the net gets at most k tokens for all markings in the reachability set $R(m_0)$.
A Petri net is safe if it is k -bounded and $k = 1$.
- Liveness - A Petri net is live with respect to a marking m_0 if, for any marking in the reachability set $R(m_0)$, it is possible to ultimately fire any transition in the net. Liveness guarantees the absence of deadlocks.
- Reversibility - A Petri net is reversible or proper if for every $m \in R(m_0)$, $m_0 \in R(m)$. Therefore, the initial marking is reachable from all reachable markings. Reversibility implies that the model can reinitialize itself. This is important for the automatic recovery from errors and failures.
- Persistence - A Petri net is persistent if, for any marking in $R(m_0)$, an enabled transition can be disabled only by its own firing. This implies that a net representing resource sharing conflicts is not persistent since the same place is an input to more than one transition. Hence, an enabled transition can be disabled by the firing of another transition.

- **Conservative** - A Petri net is conservative if its number of tokens remains constant.

Let $m' \in R(m_0)$,

$$\sum_{p_i \in P} m'(p_i) = \sum_{p_i \in P} m_0(p_i).$$

The properties of liveness, boundedness, and reversibility are independent. Liveness and boundedness are two properties that are particularly important for the application of Petri nets to system design and analysis.

Other Process Modeling Methods

System Modeling Methods

Panko (1984) developed the Strategic Approach to determine the support needs of individual offices. This technique identifies goals and functions of the office by using the Critical Success Factors Method (Rockart 1979) and/or the External Contribution Approach (Drucker 1967). Harris and Brightmand (1985) proposed The Critical Task Method for need assessment evaluation. It is assumed that support requirements are best prescribed by identifying the “bottleneck cognitive tasks” in an office. However it does not make any specific assertions on whether a particular task is susceptible to automation or assistance (Ang, Vanneste, and Lu, 1998). The Office Automation Group at MIT has developed MIT’s Office Analysis Methodologies to increase understanding office work and to specify this knowledge procedurally in order to support the process of developing office system. However the methodology worked well in only routine procedure handling

(Panko 1984). The application is limited by its lack of prescriptive guidelines; its focus is purely descriptive (Sutherland and Sirbu,1983). Sasso (1985) developed the Task Analysis Methodology. It considers office work as sets of information-handling activities and describes it using four basic concepts: information-objects, tasks, conditions, and structures of tasks. Once these descriptions have been completed, it proceeds to the analytical state that identifies opportunities for improving an office's task configuration in an office. Ishiguro and Tarumi (1997) presented the Agent-based Office System that supports both personal and group work. The agents were mutually connected in a two-layered agent network to work in cooperation. With the support of these agents, people in an office can expect to perform both their personal and group work efficiently. Ang and Conrath (1996) proposed Operations-Oriented Methodology with the objectives of being comprehensive in terms of scale and detail. It uses Petri nets and Playscript formats (a step by step notation on the net as to who does what at each node) to describe the system, with the net to be capable of formal analysis, and the Playscript to be understood by a non-expert. Yang and Ho (1996) introduced Cooperative Distributed Problem-Solving Framework this framework that was based on an Object-Oriented Cooperative Distributed Problem-Solving Shell. It used distributed Artificial intelligence and emphasizing the human agents as an integral part of the performance of required official activities.

There are other methodologies for system modeling and analysis. Most of them have difficulty describing and analyzing the complexity of office work. Various techniques of structured analysis, including those based on the object-oriented paradigm (Henderson-Sellers and Constantine, 1990), focus principally on the analysis and design

of structured and semi-structure activities, and are used essentially for systems specification. At the other extreme, the Critical Task Method proposed by Harris and Brightman (1985) emphasized unstructured (or ill-structured) work. Office Analysis and Diagnosis Methodology suggested by Sutherland (1983) is targeted at semi-structured office activities, and by self-admission it does not address the issue of understanding managerial work.

Data Modeling Methods

There are various data modeling and analysis methods for office automation system, each having its own merits and limitations (Langer 1997). 1) *A Data Flow Diagram* is a tool that shows how data enter and leave a particular process. It consists of Process, Data Flow, Data Store, and External. A Data Flow Diagram depicts flows and boundaries. It represents a graphical display of the process, and therefore is a usable document that can be shown to both users and programmers. 2) *Process Specifications* is a detailed and regimented process specification in pseudo-code or structured English. It requires the designer to have a solid understanding of algorithms. 3) *State Transition Diagram* is designed to model events that are time-dependent in behavior. A state Transition Diagram models the application alternatives for event-driven activities or any activity that depends directly on the behavior of a pre-condition that makes that event either possible or attractive. It is very effective in artificial intelligence and object-oriented analysis. 4) *Entity Relationship (E-R) Diagram* is a model that depicts the relationships among the stored data (Korth and Silberschatz, 1997). It is the most widely

used of all modeling tools and is the only method to reflect the system “stored data”. An entity in database design is an object of interest about which data can be collected. In a retail database application, customers, products, and suppliers might be entities. An entity can subsume a number of attributes: product attributes might be color, size, and price; customer attributes might include name, address, and credit rating. E-R Diagrams is very effective in modeling and analysis of finite structure.

Organizational Aspects of Office Automation

Implementing an office automation system within an organization is a complex task, usually resulting in both technological and organizational change. Many office automation projects have failed because the management and system designers have focused too much attention on technical and economic objectives while under-rating social, and individual requirements (Otway and Peltu, 1983). Paddock (1984) argued that changes to the hardware, software, data, and personnel components of an organization tend to be more noticeable than changes to the organization, planning, and control attributes that permit these components to function as a unit. Thus effective office automation planning is vital for successful operations of an automated office (Benjamin 1982, Connell 1978). The use and success of an office automation system depend largely on the attributes of the office in addition to technical aspect of the system. These attributes are: the readiness of the office, managerial support, and previous experience with office automation (Dumais, Kraut, and Koch 1988).

CHAPTER III

A CASE STUDY AUTOMATING THE MANAGERIAL AND FUNCTIONAL OPERATIONS OF A&S EXTENSION OFFICE, OSU

Extension Office Characteristics

The Arts and Sciences Extension office (A&S Ext) is a functional unit of both the College of Arts & Sciences and the University Extension of Oklahoma State University (OSU). It offers a line of credit courses (international, out-of-state, weekend, condensed, internet, compressed video), and a line of non-credit programs (academic conferences, workshops, seminars, satellite courses) for OSU. Its clients are on-campus and off-campus students, Stillwater residents, high school districts in the Mid-West region, senior citizens from all over the USA, and employees of corporations. It is a financially self-sufficient unit in terms of the courses/programs that run through the Extension, while the staff salaries in the office are funded by state allocations.

All of A&S Ext. courses/programs go through several stages before they are completed at the end of each semester. These stages are:

- Marketing stage
- Planning stage
- Course/Program preparation stage
- Teaching/Learning stage
- Program closing stage

- Reporting and revenue distribution stage.

Procedure Analysis of Basic Functional Units

Figure 2 outlines the basic functional units of A&S Extension. An “m” above the function indicates that the unit was operated manually. An “Exl” indicates that the function is operated through the use of Excel spreadsheets. A p/a indicates partial automation of the function or a combination of software package and manual operation. No symbol indicates that the function did not exist before the integrated automation system.

Below is a detailed analysis of the operational procedures for each of the functional units:

Marketing

The tasks and their sequence are 1) Directors/Coordinators meet with departmental faculties to discuss ideas of offering new/existing courses/programs or conferences. 2) The marketing team reaches a preliminary agreement on program title, dates, location, contents, text book, necessary materials, marketing method, delivery, incomes, expenses, payments, as well as promotion strategy. 3) Each coordinator in charge of the program designs promotional pamphlet/fliers for the program. 4) Coordinator iterates several times with the instructor and the director about the promotional material. 5) Finalize the promotional material and present it to the public. 6) The support staff of the course/program prepares mailing list and letters.

A Case Study – Arts & Sciences Extension Office

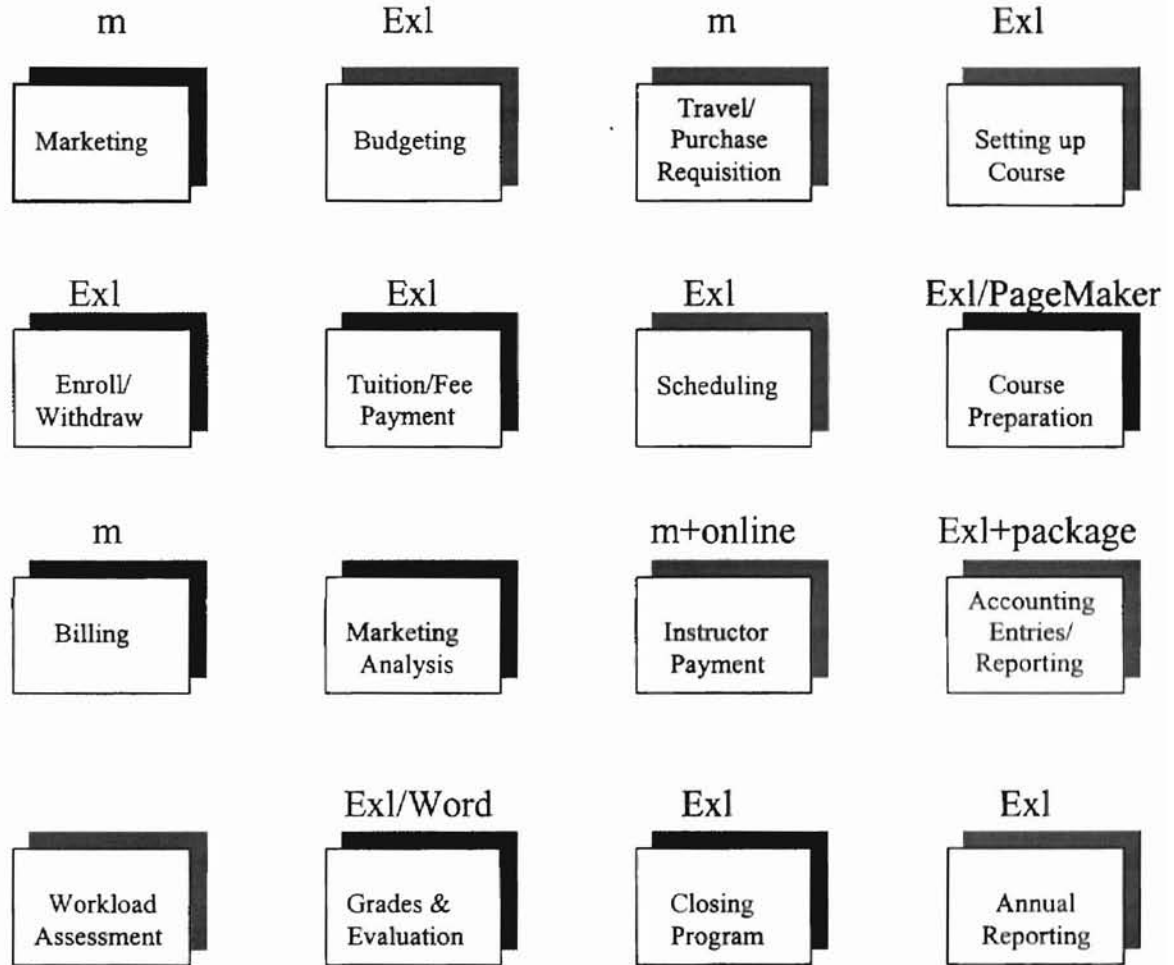


Figure 3. The basic functional units of A&S Extension office

Budgeting

The tasks and their sequence are 1) The coordinator prepares a budget draft based on the preliminary discussion with the department, instructor(s), and Extension Directors. 2) The draft is sent out to the department and the instructor for their inputs. 3) After several iterations the department and the instructor(s) sign the budget. 4) The budget along with course/program information is sent to the College and University Extension for approval. 5) The budget is either approved or disapproved by the College and University Extension.

Preparing Purchase Orders and Travel Requisitions

If a course/program has an expense that is expected to exceed OSU small purchase limit of \$1,500, then a purchase request must be prepared. Similarly, if an instructor must travel out of Oklahoma for teaching, a travel request for the instructor must be prepared. As it takes three weeks to complete the paper work, when time is restricted the purchase order and the travel request are sent out regardless of the approval state of the program. The course/program can be canceled if it is not approved. The Purchase order and the travel requisition are prepared by typing the information onto the forms with a typewriter.

Setting up Program

The tasks and their sequence are 1) Receive the budget draft from the coordinators, 2) Assign a program ID to each new course/program. 3) Enter course/program information onto the system. 4) Based on the budget draft information,

enter the tuition/fee structure and refund policy for each program ID. 5) Run an Extension program status report periodically. 6) Revise the Tuition/Fee structure and refund policy base on the budget revisions. This process could take several iterations.

Accounting Entries and Reconciles

A program ID is also its account number under Extension office major accounts. Once a program is set up, incoming expense/revenue can be posted into its program account. Thus the tasks and their sequence are 1) Confirm that a program ID has been assigned to the program. 2) Posting expenses/revenue of the program. 3) Correct errors by entering a reversed posting. 4) Run preliminary program statement. 5) Get feedback from the coordinators and adjust the postings. 6) Run final income statement for the program. 7) Program coordinator signs the statement.

Monthly accounts must reconcile with University account statements. This procedure is performed monthly on all Extension accounts to ensure the correctness of all the entries in the Extension Office. The tasks and their sequence are 1) Print monthly Extension account reports. 2) Check the reports with University account statements, mark any expenses/revenues that are not recorded in University statements. 3) Correct any errors by looking into the original invoice/check copies. 4) Generate a “reconcile statement” for the account. 5) Repeat 1)-4) until all accounts are reconciled.

Taking Enrollment/Payment

The Extension Office takes the enrollment if the course is not entitled to enroll through the University Sectioning. The procedures and their sequence are 1) Enroll the student/participant to the program. 2) Enter student/participant's information if he/she is enrolling in Extension program for first time. 3) Record the payment if the tuition/fees is paid during the enrollment. 4) Record the withdrawal of any participants from a program. 5) Checking the refund policy to decide whether a refund is necessary. 6) Refund if necessary.

Scheduling Program Activities

The support staff of the program does a sequence of tasks. 1) Enter deadlines to the program, such as mailing dates, text book order date, participant list date, final grade turn-in date, etc. 2) Make notes about the special requirements for the program. 3) Run a to-do-list report either for a single program or for all the programs the support staff is working on.

Preparing Program Materials

The support staff for the program does all or some of following tasks before the program starts. The tasks can be done in parallel. 1) Set up a program file in the office. 2) Verify the total enrollment for the program. 3) Prepare participant list with name and address on it. 4) Prepare a sign-in sheet for some programs. 5) Prepare nametags for conferences. 6) Prepare mailing list for the confirmation letters. 7) Order and receive textbooks for the program. 8) Prepare handouts or packages.

Billing

The sequence of billing procedures is 1) Generate a payment report for each program. 2) Remove any one who will be billed through OSU campus vendor invoice (CVI) from the list. This can be done by preparing a CVI and enter as a payment for the participant. 3) Prepare bills for each participant on the list. 4) Record the bills as payment for each participant into the program. 5) Send the bill to Bursar's office and the participant.

Marketing Analysis

A market analysis report will be prepared to show the effectiveness of each promotional medium. The most effective promotional method for the program will be identified. The procedures are 1) During the enrollment, record the promotional medium by which each participant is made aware of the program. 2) Calculate the percentage and print the results for each program as well as for all program types.

Teaching/Learning

Teaching and learning activities proceed as scheduled. Location is on-campus, off-campus, or international. Delivery media will be classroom, two-way compressed video, regular video, satellite, and/or internet. The Extension office is responsible for the printing of all handouts, homework assignments, and exams in a timely manner.

Instructor and Teaching Assistant (TA) Pay

The procedure for instructor pay is done as follows. 1) Confirm the ending of the course/program. 2) Verify the amount with program coordinator. 2) Verify pay type and pay dates with the Dean's office. 3) Call department to update instructor's EA if it is to be an inload payment, or prepare a special pay form in the Extension office if it is to be an overload payment. 4) Approve the payment by the Extension Director and the Dean's office. 5) Send the paper work to OSU payroll department for processing. 6) Enter the payment along with actual benefits amount into the program account after receiving monthly payroll reports from University Accounting department.

Sending out Grades and Evaluations

For the grade report, 1) Confirm the grades with the instructor. 2) Send the grades to the University Administration. For course evaluations, 1) Collect the evaluation forms. 2) Enter the evaluations into the system. 3) Send the evaluations to the department.

Closing A Program

The program accounts are closed at the end of each fiscal year. No changes are allowed to the account thereafter. The procedures are 1) Update the program income statement. 2) Move program income/deficit into an office control account. 3) Lock the program account.

Distributing Income

The net income of the program will be distributed to the participating departments according to the budget agreements. The procedures are 1) Determine the distribution percentage per budget agreement. 2) Calculate the amount for the departments. 3) Transfer the funds into departmental accounts. 3) Send statements to the departments.

Annual Reporting

The annual reporting of the Extension office consists of the following reports not necessarily in sequence. 1) Summary information on all courses/programs in terms of total course/program offered, total credit hours offered, total gross income, total program expenses, total office expenses, total net income, total enrollment, total instructor participation, total department participation, total grant/sponsorship received, total payment for instruction. 2) Detailed information on each course/program offered, sorted by program types, by departments, and by coordinator in charge of the programs. 3) Detailed information on instructors regarding the courses/programs taught, total credit hours and enrollment, and total inload/overload payments. 4) Total on-campus expenses.

Pre-Integrated Automation

Before the integrated automation system was introduced in 1998, the staff in the Extension office had been using office automation tools such as Microsoft Excel, Word, PageMaker, and a DOS driven registration and accounting package developed in 1986 by KM Register Inc., a small consulting firm, to handle day-to-day operations. However, it

was not a coherent and integrated system. Each staff member operated on her/his own PC or MAC, and created her/his own data files that were not shared with other staff members. The result was a classical example of many "isolated operational islands" in the office. The system suffered data redundancy, inconsistency, conflicts, and repetition. It was inefficient and unproductive.

During a conference period, preparing handouts, name tags, and billings involved slow, labor intensive procedures with duplication of effort in various functional units. For example, the same registration and tuition payment information would be keyed in at least three times during registration, payment/deposit processing, and accounting entries, respectively. Errors were inevitable. Several times a student had dropped a program, while the system showed that he/she was still listed on the program. Instructors' pay checks were sometimes delayed. Programs with the same title but different sections would be either confused altogether, or the expenses were mis-posted. In addition, each coordinator planned his/her own program/course schedule without knowing the upcoming events planned by the other coordinators. This sometimes caused bottlenecks for the office, and almost always slowed down operations.

In spring 1997, The existing registration/accounting software crashed. It had been upgraded twice during the preceding ten years to bring it up from a stand-alone application into a network environment application. The firm stopped supporting the software shortly after the second upgrade. The Extension Office thus had to decide whether to repair the package or to find a new office automation application that could meet its daily operational needs.

The Decision to Automate and Process Reengineering

The Extension Office made the decision that a new system was imperative in spring 1997. A committee was assembled to search for a new system. The question of selecting either an off-shelf software package or a customer-designed system was addressed. After a period of intensive search, various software packages available on the market were tried and tested. Consulting firms specializing in software development were interviewed. It was clear after the search that the existing off-shelf software could not meet the operational specifications of Extension Office. A customized workflow redesign is the solution to integrate and streamline the basic functional units of Extension Office.

To implement a new integrated system fully, the office was ready to redesign its existing work processes as well as to change the job descriptions of certain staff in the office. The “old ways of doing the things” would be changed. The office would adopt a new, more efficient operating procedure to perform its day-to-day operations. The issue of change was brought up in a multitude of staff meetings. Most of the staff showed enthusiasm about the upcoming new system, eager to switch to the new system when it was ready and operational.

The redesigned workflow diagram for the Extension Office is shown in figure 3. It is a diagram representation of the processes of running a single extension program. It streamlines all the processes/tasks of a program. Some of the processes must run in sequence, while others can go in parallel. The workflow begins with the marketing process, followed by the budgeting and the program set up processes. The travel requisition and/or purchasing order process starts as soon as the budget information is

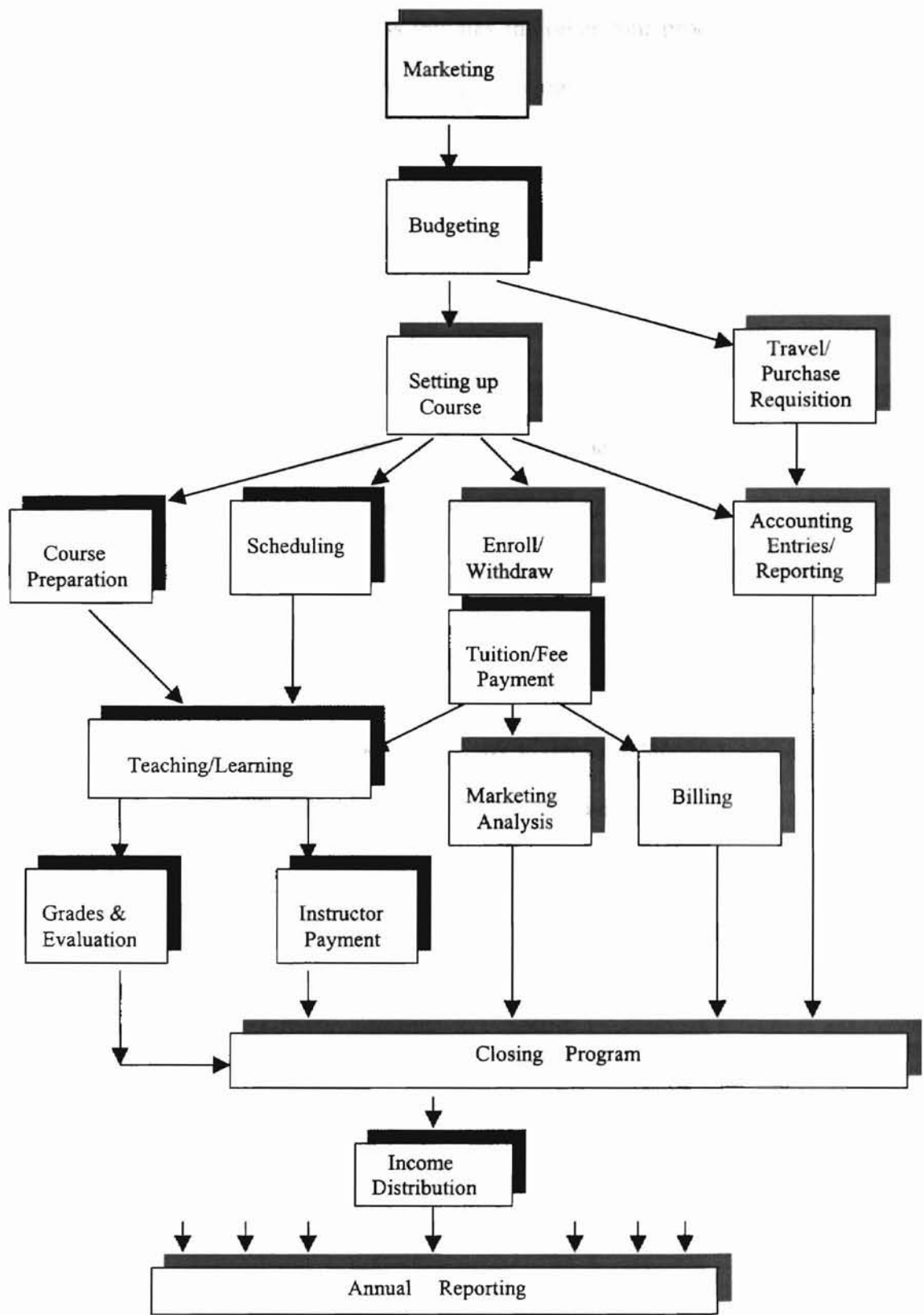


Figure 4. Extension Office Workflow Diagram

available. The program set-up process initiates the other four processes simultaneously. The support staff begins to make a file for the program in the office. All the information about the program is put into the file. The schedule for the program is listed. The front desk starts taking enrollment for the course/program. Any invoices for the program can be recorded into the program account. After the course preparation and the scheduling processes is the teaching/learning stage. Once it is completed, grades are reported, instructor and TA's are paid. The billing process operates in parallel with the teaching process. The program closing process starts when all expenses and tuition/fees have been received and recorded (posted into program account). At the end of the workflow is the annual reporting process. The program is listed on the annual report after its closing.

The workflow repeats itself for each course/program that runs through the Extension office. The annual report is the result of the reports of all the programs that operate through the Extension workflow. At any given time there are about forty (40) to fifty (50) programs running at the Extension office. Thus, there are about forty to fifty course/programs going through the processes simultaneously in workflow.

As a result, some staff's duties or job description, and even the physical work locations must change in order to meet the requirements of the automation system. Data redundancy, repetition, inconsistency, and conflict can be eliminated since data is now shared among the office staff, and is updated constantly. As each staff member in the workflow only accesses the part of data files that is relevant to his/her job description, data security is assured.

Design Objectives

As Law and Gorla (1996) stated that the success of an office automation project was directly related to the degree of its integration into everyday business tasks. A user friendly interface results in more user satisfaction, and hence, more use. Thus system integration and user friendliness are the two major design features for the Extension office automation. Below are the objectives of the system for the Extension office:

- 1) Integrate all the existing as well as the new functional units in the office.
- 2) Automate about 80% of functional units in order to increase the productivity and efficiency.
- 3) The software should be PC and MAC computer compatible since the office has ten (10) PCs and six (6) MACs.
- 4) Provide network capability in order to access external databases.
- 5) Exchange information with existing software tools such as Microsoft Office, PageMaker, etc.
- 6) The software should be user friendly, easy to learn and use.
- 7) The software should be easy to maintain and upgrade
- 8) The initial phase, automating Budgeting, Program Management, Registration, and Financial Accounting, should be developed within the budget limit of \$10,000.
- 9) The enhancement phase should include rest of the functional units, with a new budget agreement to be discussed after the completion of phase I.

Workflow Redesign Using Petri Net

General Model of Extension Office

Figure 4 is the Petri net model for the Extension office workflow. The textual representation of the model is shown in Appendix A. The tokens in the net represent the information of the program. The set of transitions T_i represent operations, such as information processing, communication, which take place at various stages of the workflow in the office. The set of places P_i represent conditions or states in the workflow, although they can be perceived as inputs or outputs of various transitions.

Since the basic properties of the general Petri net are well investigated and have a firm mathematical foundation, the workflow model for the Extension office uses the general Petri net at higher levels for analysis. To model the details of the functional units as well as to avoid the models becoming too large, the hierarchy constructs are used. For each construct, there is a corresponding subnet describing the specific details of the functional unit. There are two(2) levels of hierarchy used in the Extension office model.

The initial number of tokens in the net is one(1), or one program is running at the time of analysis. The weight of the arc is one(1) for all the arcs in the model. As the transitions fire one after the other, the token(s) flow through the net, which represent a program is being initiated, prepared, ran and completed. The same life cycle is repeated for each program that run through the Extension office.

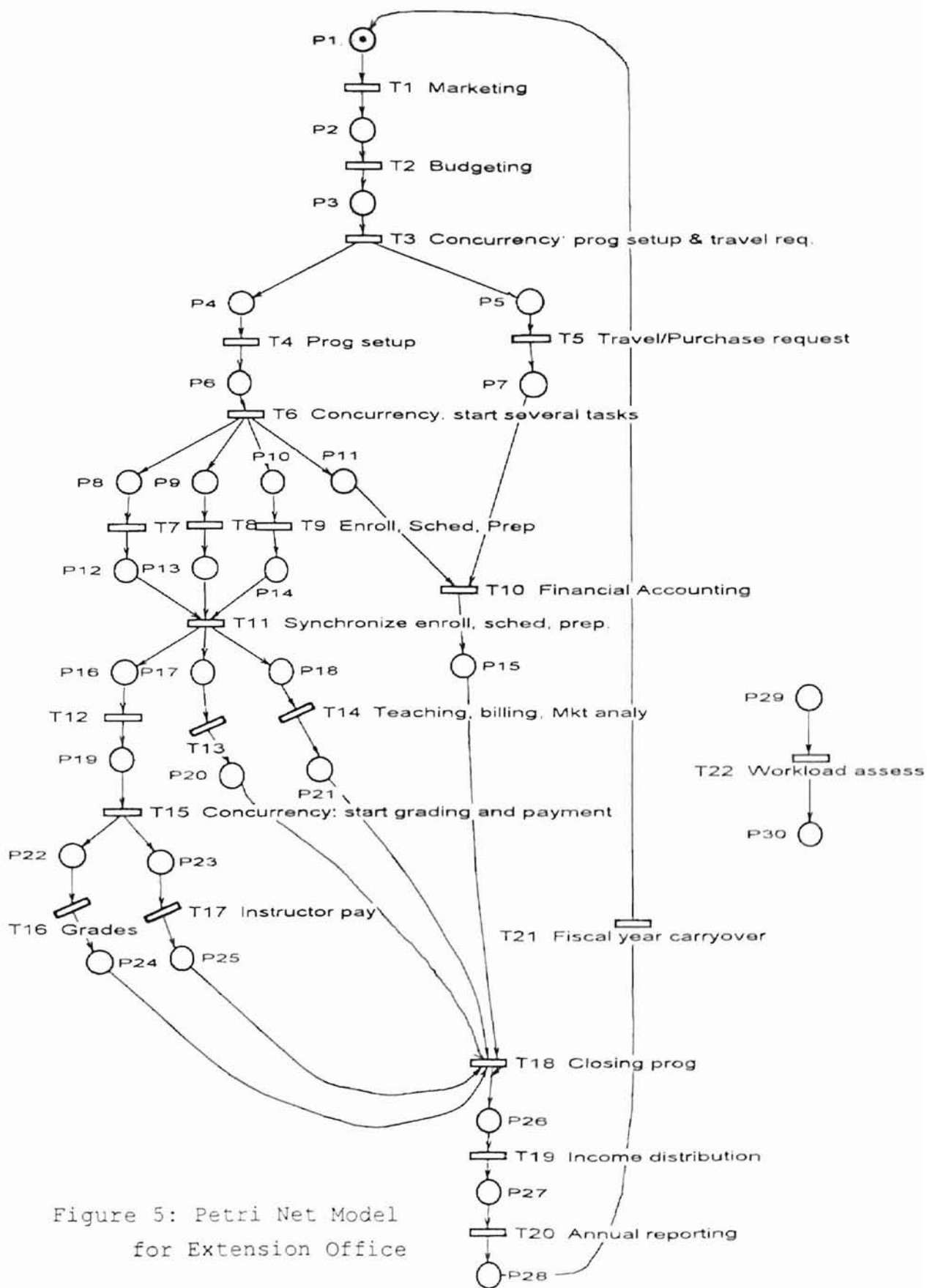


Figure 5: Petri Net Model for Extension Office

Below are the definitions for each place P_i and the transition T_i in the workflow model.

Places:

P_1 – Input or new idea is available on the development of Ext. programs. For example, some faculties showed interests on teaching ext courses, departments want to sponsor an academic conference, or workshop, etc.

P_2 - program information is advertised on the media (fliers, newspaper, internet, TV, etc.)

P_3 - program budget draft is prepared by the coordinator

P_4 – receptionist received program budget information from coordinators

P_5 – purchasing/travel information is received by Ext. office

P_6 – receptionist assigned an ID to the program, and listed it into the system

P_7 – purchasing/travel requisitions have been approved

P_8 – received information for enrollment

P_9 – received information for program scheduling

P_{10} – received information on program preparation materials.

P_{11} – receive invoices from vendors

P_{12} – enrollment report for the program

P_{13} – To do list ready

P_{14} – program preparation materials ready

P_{15} – expenses entered into program account

P_{16} – received all necessary materials/information for teaching delivery

P_{17} – payment analysis report for billing purpose

- P₁₈ – marketing information received
- P₁₉ – teaching delivery completed
- P₂₀ – Ext. invoices generated and send out
- P₂₁ – marketing analysis report ready
- P₂₂ - received grading and evaluation information
- P₂₃ – received instructor & TA payment information
- P₂₄ – grade and evaluation put into program file
- P₂₅ – sent payment information to payroll department
- P₂₆ – program income statement
- P₂₇– program income distributed into Ext. and departmental accounts
- P₂₈ – annual report completed
- P₂₉ – start enter information on workload.
- P₃₀ – workload reports

Transitions:

- T₁ – preparing marketing material, recording activities (meetings, interviews, etc)
- T₂ – finalize and prepare a budget
- T₃ – send budget information into program and accounting module
- T₄ – assign program ID, setup program fee structure for enrollment
- T₅ – prepare requisition for purchasing/travel
- T₆ – concurrency: starts accounting, registration, program mgmt module
- T₇ – taking enrollment, receiving payments, processing refund
- T₈ – setup deadlines for various activities of the program

- T₉ – preparing confirmation letter, sign in sheet, participant list, name tags
- T₁₀ – financial accounting entries on program expenses/incomes
- T₁₁ – synchronizing enrollment, scheduling and program preparation
- T₁₂ – teaching activities
- T₁₃ – run billing invoice
- T₁₄ – marketing analysis
- T₁₅ – concurrency: send information on payment and grading
- T₁₆ – grading and evaluation
- T₁₇ – prepare payment form
- T₁₈ – synchronizing accounting, billing information. Closing account
- T₁₉ – distribute incomes
- T₂₀ – prepare annual reports
- T₂₁ – fiscal year carryover. Update system categories and subcode. Adding policy changes into the system.
- T₂₂ – workload assessment. It is running parallel with the main system.

Marketing Decomposition

The marketing Module includes transition T_1 Marketing, and transition T_{13} Marketing analysis reporting. T_1 can be further decomposed into the initial discussion with the instructor/head of department/sponsor about the course/program, and the finalization. Several iterations are needed before the final agreement on the course and the advertising content can be reached.

Figure 6 shows the decomposition of transition T_1 .

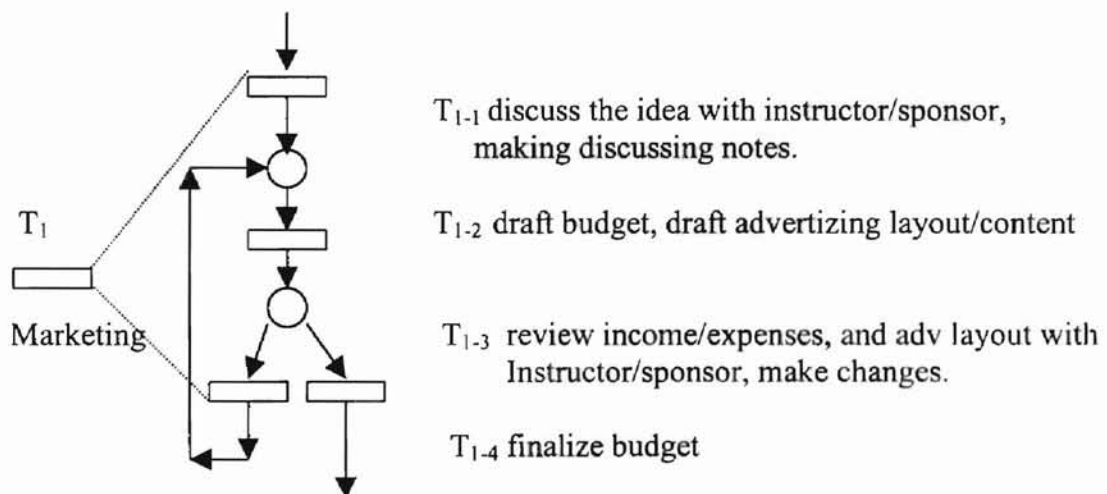


Figure 6. Sub-net of marketing

Planning and Budgeting Decomposition

Planning and budget transition T_2 involves several plays of “What if” analysis on the budget platform. The coordinator enters the estimated numbers of attendant and the tuition/fee charged for the course. The total revenue, total planning expenses, fixed expenses, variable expenses are calculated. The administration fee for the program is calculated as well according to the percentage rate selected by the coordinator. The program net income then is displayed, and the breakeven number or the minimum number of attendees for the course to run is displayed. If the course loses money, the net income will show red. The coordinator can change the attendance figures or the expense amount to satisfy the breakeven requirement. Figure 7 shows the decomposition of T_2 .

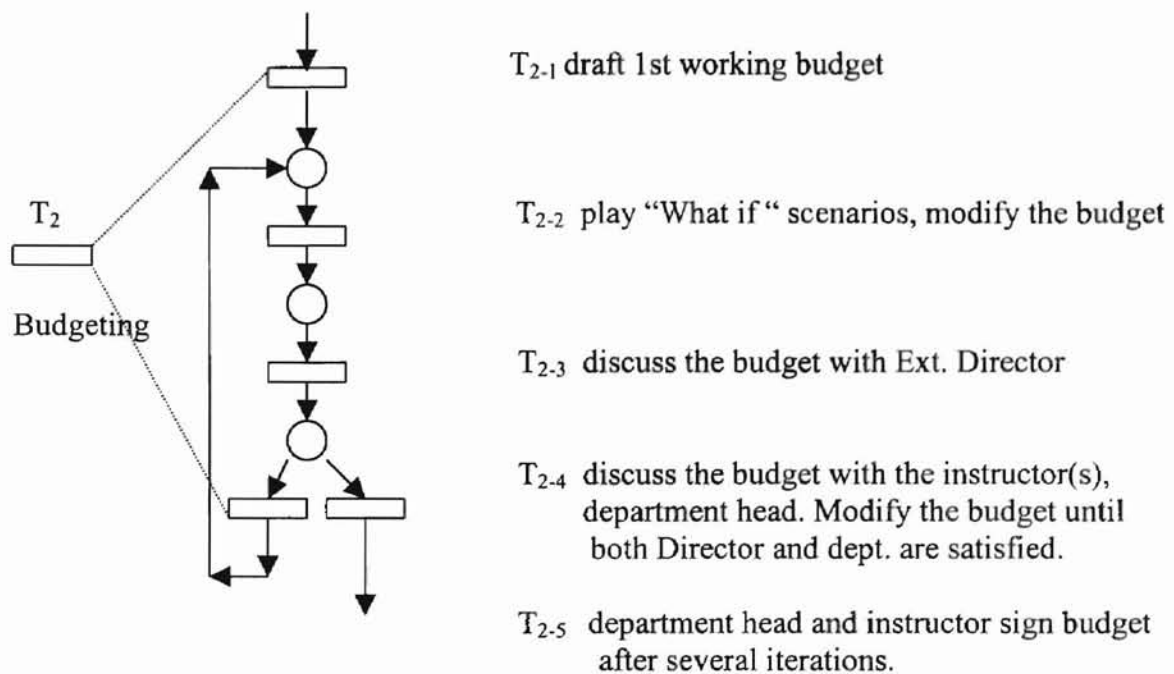


Figure 7. Sub-net of budgeting

Registration Decomposition

The registration transition T_7 consists of four basic functional units: Take Enrollment, Receive Payment, and Refund for those cancelled before the course deadline. Thus T_7 can be further decomposed as show in figure 8:

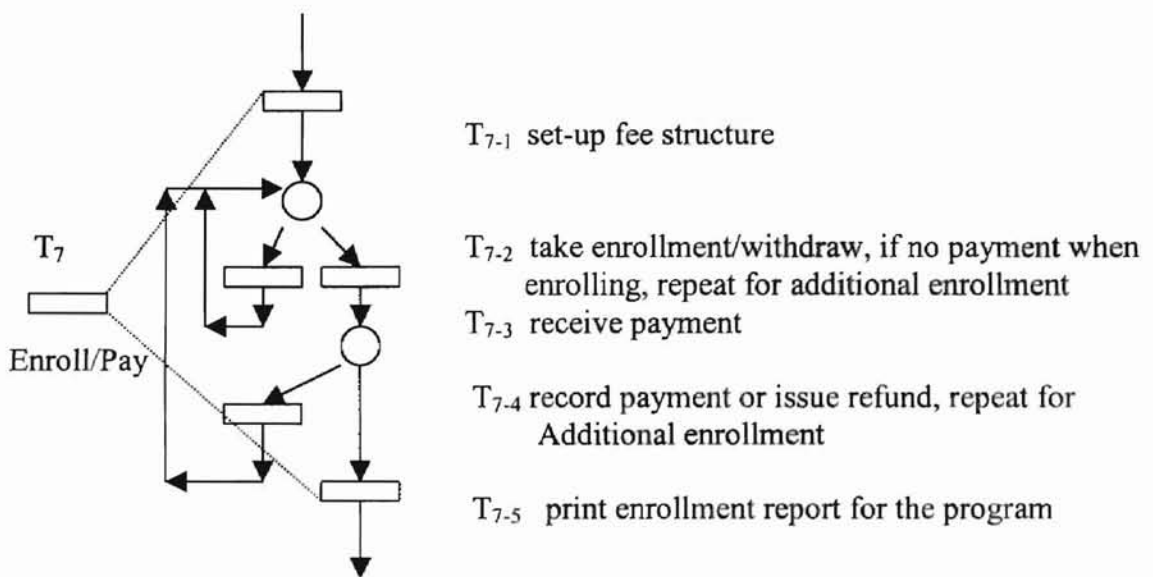


Figure 8. Sub-net of enrollment/payment

Program Preparation and Billing Decompositions

Program preparation is part of the program management module that includes T₄ program fee setting up, T₉ Prepare course materials and run course reports, T₈ Scheduling, and T₁₃ Billing. T₉ Prepare course materials and run course reports can be further refined as shown in fig 9. T₁₃ Billing can be refined as shown in figure 10.

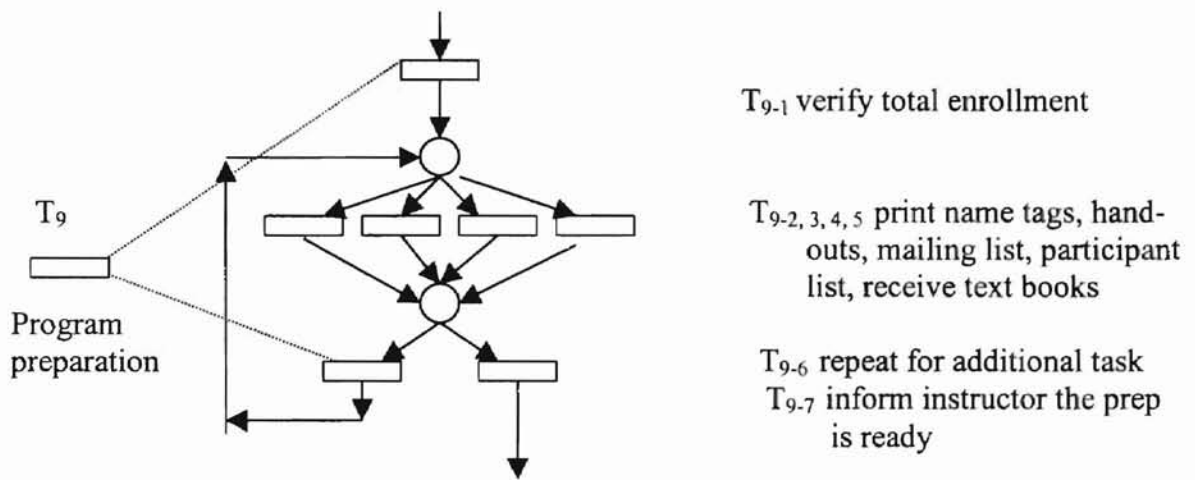


Figure 9. Sub-net of program preparation

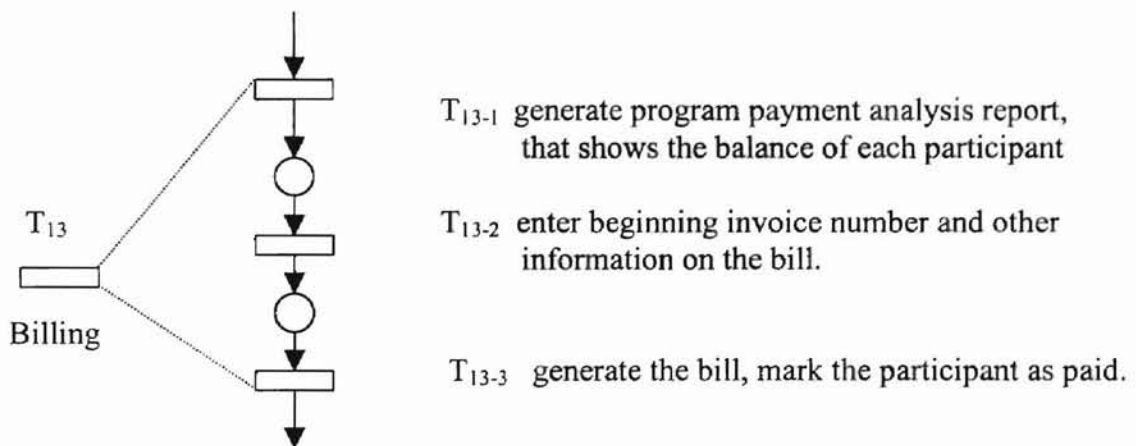


Figure 10. Sub-net of billing

Financial Accounting Decomposition

The financial Accounting module manages the financial and accounting activities of the Extension Office. Its functions include 1) accounts set-up, 2) income/expenses posting, 3) monthly accounts reconciliation with University Accounting Statements, and 4) end of year account balance carryovers. Various categories and sub-codes are set up to meet the accounting requirements of State Agencies. Standard fee/tuition policies are updated and modified regularly. Various procedures and deadlines must be followed to ensure the correctness of the accounting system. Figure 11 is the decomposition of T_{10} Financial Accounting transactions.

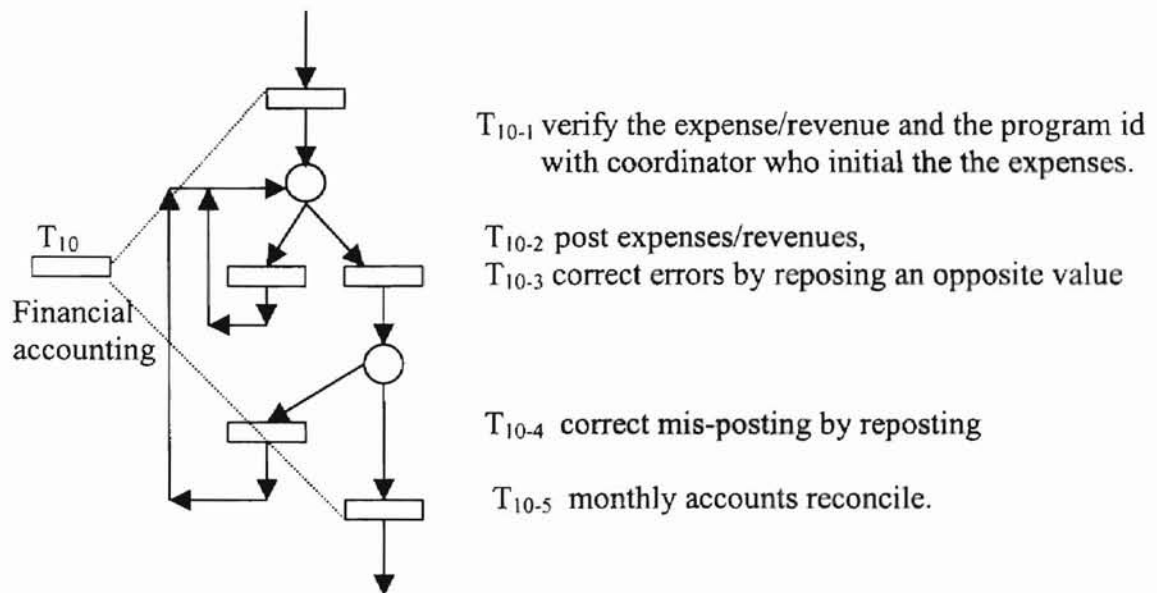


Figure 11. Sub-net of financial accounting

Managerial Decision Making Decomposition

To help the Directors of the Extension Office evaluating the courses/programs, a Managerial Decision Making module is set up to fulfill this requirement. One of the component in the module is Workload Assessment T_{22} . It includes entering the hours spent on the program, and generates workload assessment report on each program as well as on each coordinators. It is running independently, and is in parallel with the main system. It is a part of the system.

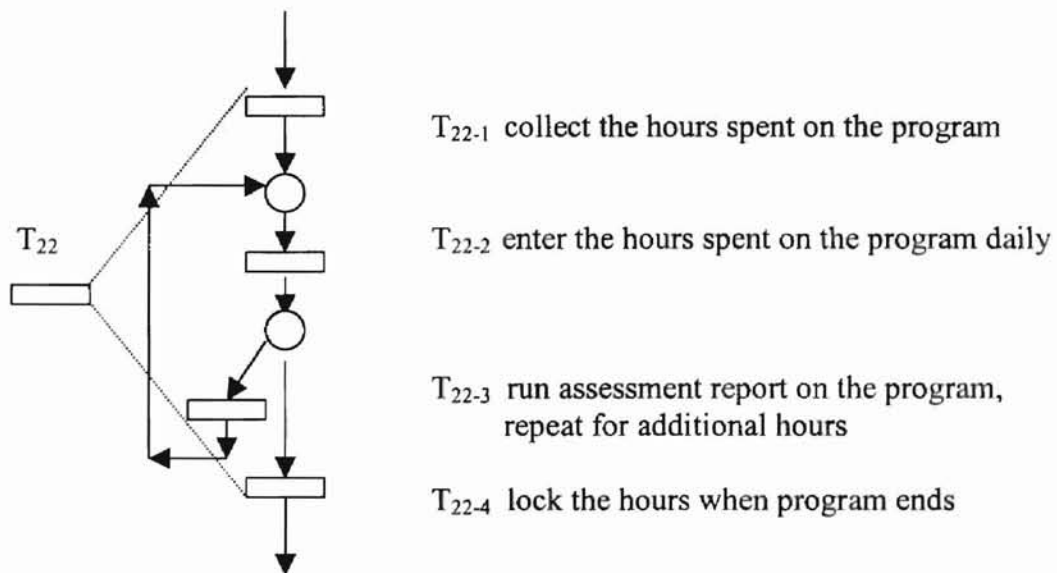


Figure 12. Sub-net of workload assessment

The Architecture of Automation

The architecture of Extension office automation is shown in fig 12.

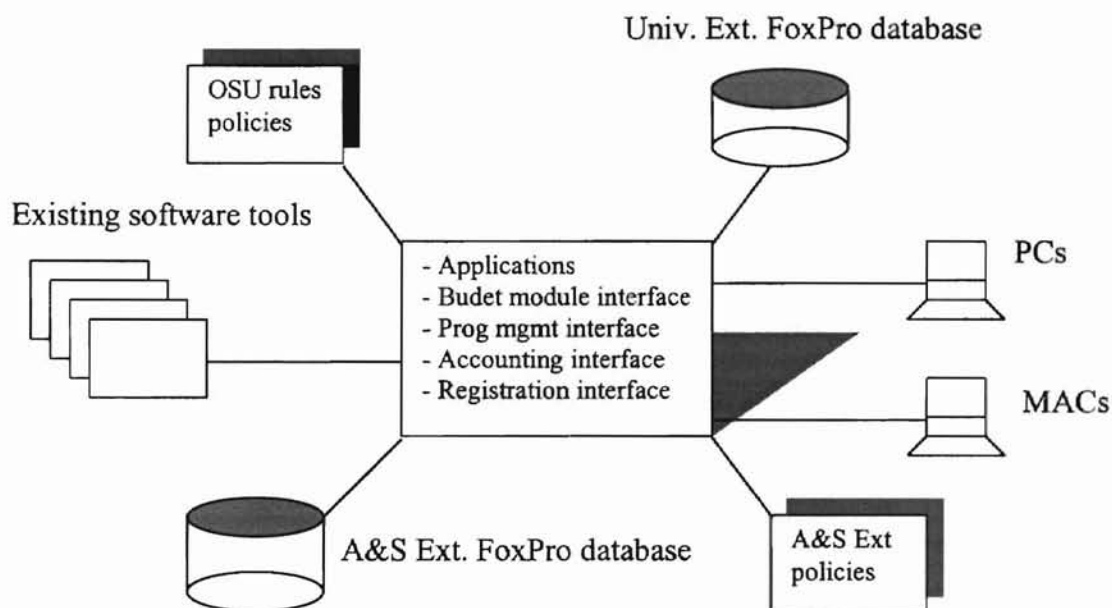


Figure 13. Architecture of A&S Ext. office automation

Existing Software Tools

The office had been using several software tools, such as Microsoft Exel 97, Word 97, PowerPoint 97, Macworld PageMaker 6.5, and Adobe PageMill 2.0 etc., to perform its day-to-day operations prier to the new workflow system. The new system is linked with the existing tools through the data import and export facilities. Thus the

powerful functions of the existing software tools can be utilized fully and become part of the automation system.

FoxPro Database

FoxPro 3.0 is used as the database development tool for the Extension automation system. Since FoxPro 3.0 is the only database development tool available on the market that can work on the platforms of both PCs and MACs, it is selected to accommodate the needs of both PC users and MAC users in the Extension office. A later version of FoxPro 6.0 runs much faster on PCs, but it does not run as well on MACs as the version 3.0 does. For version 3.0, due to the transfer of the platform, the applications on MACs are run much slower compared with the applications run on PCs.

PCs and MACs

The Extension office had ten (10) PCs and six (6) MACs at the time of the workflow development. Directors and the program coordinators were using the MACs, the rest of the staff and student workers were using PCs. There was no immediately sign that the MAC users would replaced their machines with the PCs.

External Databases

The University Extension (UE) was developing its own database management system using the Foxpro 3.0 database development tool at the same time as the Arts & Sciences Extension office was developing its own automation system. The UE has been requesting its annual reporting information from the Extension offices across the campus. It was done through collecting the forms and through re-entering the program information via the campus network into an user interface specially created for the purpose of collecting data.

The new system is able to connect directly with the UE database. It sends the information requested by the UE into their system via the network. It also can exchange information with other Extension office across the campus via the central database in the UE.

However, the new system is not directly linked with the other database systems on the campus, such as University Accounting database, and the Payroll database. These systems are managed by Computer Information Services (CIS), and the Extension offices are not allowed to access them.

CHAPTER IV

WORKFLOW VALIDATION AND VERIFICATION

The correctness, effectiveness, and efficiency of the office processes supported by the workflow automation system are vital to the organization. A workflow automation system that contains errors leads to inefficiency, backlogs, loss of customers and loss of goodwill. Thus, it is important to analyze a workflow automation system before it is put into use in the office. One of the analysis methods is validation that tests whether the system behaves as expected. The other is verification that establishes the correctness of a workflow automation system (Van Der Aalst 1998). Performance analysis evaluates the ability of the system in meeting the requirements with respect to throughput times, service levels, and resource utilization. Since performance analysis is more useful to an industry automation system than to an office automation system, we focus our analysis on the simulation and verification aspects of the Extension office workflow model.

A software tool PESIM for Petri nets drawing, simulation and analysis (Skacel 1992) was used to perform the system simulation and analysis of the Extension workflow. The type of Petri nets that can be simulated and analyzed by this tool are general place/transition Petri nets, marked graphs (Petri nets in which each place has exactly one input and one output transition), and stochastic Petri nets (high-level Petri nets with time variables).

Validation via System Simulation

Validation can be demonstrated by interactive simulation: a number of fictitious cases are fed to the system to determine whether they are handled well. To simulate the system using the software tool PESIM, an initial marking of one (1) was placed in p_1 . A sequence of transitions $\{ t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}, t_{21} \}$ were fired to observe the movement of the token or the change of the marking in the system. Table 1 is a Flow Matrix that shows the input and output relations in terms of the transitions in the system. An integer one (1) in the matrix indicates the relationships. For example, transition t_1 has input p_1 , and output p_3 ; transition t_6 have input p_6 , and output p_8, p_9, p_{10}, p_{11} , and so forth. Table 2 is the Change Matrix that shows the changes in the markings. A negative sign indicates that a token has been removed from the place due to the firing of the transition. An integer indicates the adding of the token(s) to the place due to the firing of the transition(s). For instance, firing of t_{11} causes one (1) token to be removed from places p_{12}, p_{13}, p_{14} , respectively, and one (1) token to be added into place p_{16}, p_{17}, p_{18} , respectively, and so forth.

Several firing sequences were tested, such as $\{ t_1, t_2, t_3, t_4, t_6, t_7, t_8, t_9, t_5, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}, t_{21} \}$, $\{ t_1, t_2, t_3, t_4, t_5, t_6, t_{10}, t_9, t_7, t_8, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}, t_{21} \}$, $\{ t_1, t_2, t_3, t_5, t_4, t_6, t_{10}, t_7, t_8, t_9, t_{11}, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}, t_{21} \}$, etc. The net behaved exactly as expected in all aspects: sequential relations, concurrency and synchronizing relations, loops, etc. Thus the net can be shown to be valid in these cases.

	T1 in out	T2 in out	T3 in out	T4 in out	T5 in out	T6 in out	T7 in out	T8 in out	T9 in out	T10 in out	T11 in out	T12 in out	T13 in out	T14 in out	T15 in out	T16 in out	T17 in out	T18 in out	T19 in out	T20 in out	T21 in out		
P1	1																					1	
P2	1	1																					
P3		1	1																				
P4			1	1																			
P5			1		1																		
P6				1		1																	
P7					1														1				
P8						1	1																
P9						1		1															
P10						1			1														
P11						1				1													
P12							1				1												
P13								1				1											
P14									1			1											
P15										1										1			
P16											1	1											
P17												1		1									
P18												1			1								
P19													1			1							
P20														1							1		
P21															1						1		
P22																1	1						
P23																1		1					
P24																	1				1		
P25																		1	1				
P26																				1	1		
P27																					1	1	
P28																						1	1

Table 1. The Flow Matrix of Extension workflow net

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	
P1	-1																					1
P2	1	-1																				
P3		1	-1																			
P4			1	-1																		
P5			1		-1																	
P6				1		-1																
P7					1														-1			
P8						1	-1															
P9						1		-1														
P10						1			-1													
P11						1				-1												
P12							1				-1											
P13								1			-1											
P14									1		-1											
P15										1									-1			
P16											1	-1										
P17											1		-1									
P18											1			-1								
P19												1			-1							
P20													1						-1			
P21														1					-1			
P22															1	-1						
P23															1		-1					
P24																1		-1				
P25																	1	-1				
P26																		1	-1			
P27																			1	-1		
P28																				1	-1	

Table 2. The Change Matrix of Extension workflow net

Verification via Net Property Analysis

The correctness of the workflow is shown by the analysis of the basic properties of Petri net defined in chapter II. Linear algebraic techniques can be used to verify many properties, such as place invariants, transition invariants, and (non-)reachability. The Petri net tool PESIM was used to shown the correctness of the Extension office workflow system. The reachability tree and the markings are generated by the tool, as are the basic properties of the net under investigation.

A few adjustments were necessary before the system was put under analysis using PESIM. They are:

- 1) To avoid the models becoming too large, the analysis was performed on the general model of the Extension office workflow system, instead of on the detailed models of all the subnets in the system. The size of the reachability tree could increase without bound as the net becomes large, since it is the tree of all possible markings in the system. The number of possible markings increases geometrically with the increase of the number of places and transitions.
- 2) We are essentially interested in finite processes, or what Rozenberg (1986) calls "finite behaviors." Thus, when considering processes with possible iterations or loops, we examine only primary processes, those where individual feedback loops or cyclic components are traversed no more than once. Processes which traverse a given loop more than once are referred to as secondary processes, and there can be an infinite number of these. The limitation of considering only primary processes does not really diminish our

ability to undertake formal analysis, because it does not alter the basic properties of the systems (Gerthelot 1985).

Figure 14 through Figure 17 is the net properties of the Extension office workflow analyzed by the Petri net tool PESIM. Four (4) different methods were used by the tool to calculate the net properties. Figure 14 is the net properties calculated by the reachability tree. A reachability tree is the computation of all possible future markings starting from the initial marking. Figure 15 is calculated by the invariants of the net. P-invariant and T-invariant are nonnegative integer vectors. The total number of initial tokens in m_0 , when weighted by the P-invariant, is constant. T-invariant implies that if the firing count vector is identical to the T-invariant, then the final marking is equal to the initial marking. Figure 16 is calculated by the Marked graph. A marked graph is a Petri net in which each place has exactly one input and one output transition. Figure 17 is calculated by the Free-choice net (see Figure 19 for the informal definition of a free-choice net).

Reachability

The graphical view of the reachability tree is displayed in part in Figure 14. The complete tree can be viewed in Appendix B. The equivalence reachability tree in text form of view is shown in Appendix C. The simulation results showed that all the markings on the net can be reached from the initial marking M_0 . Thus the net is reachable.

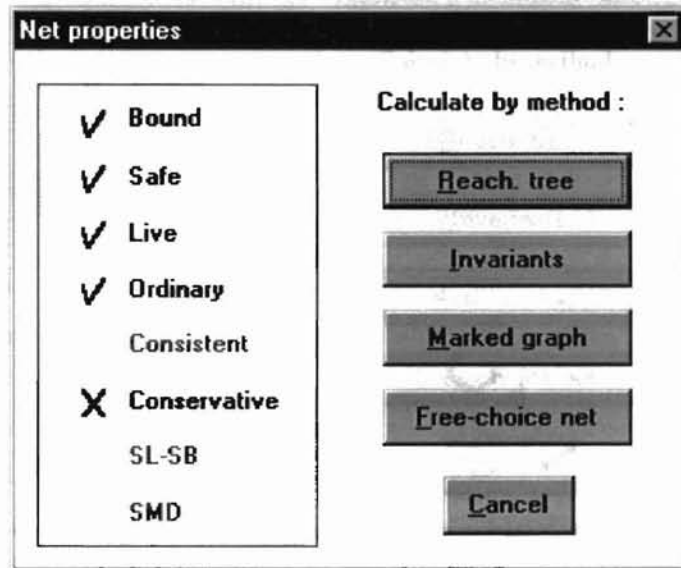


Figure 14. Net properties calculated by the Reachability Tree of the workflow

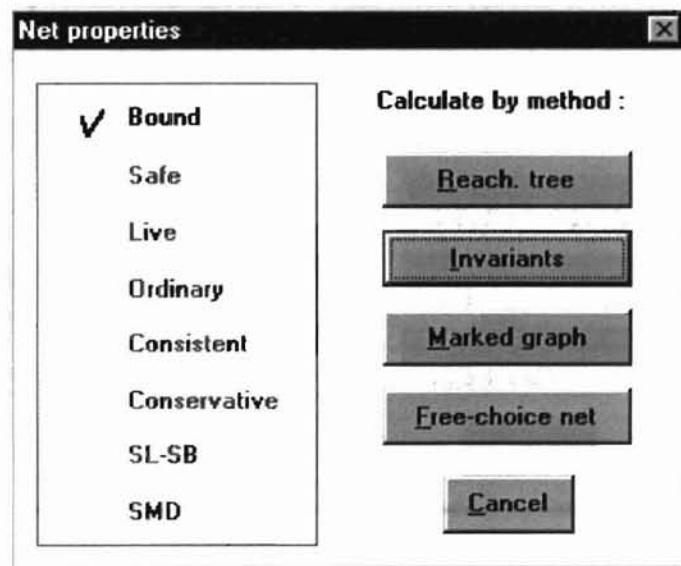


Figure 15. Net properties calculated by the Invariants of the workflow

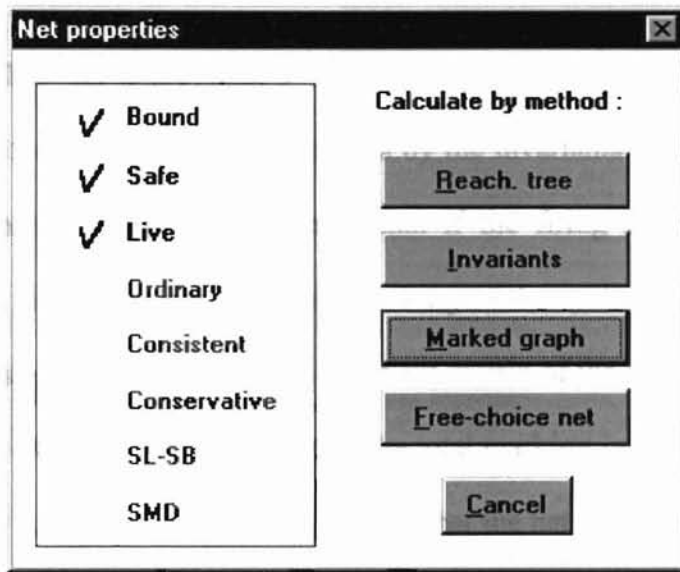


Figure 16. Net properties calculated by the Marked Graph of the workflow

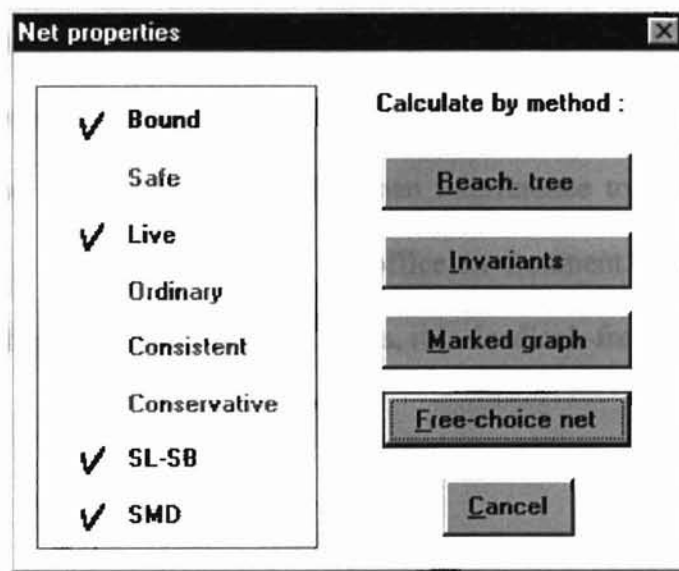


Figure 17. Net properties calculated by the Free-choice Net of the workflow

Boundedness

The workflow net is k -bound, $k=1$. By definition in chapter II, the net is bound and safe. Figure 15 calculated the boundedness by the invariants of the net. Figure 19 is the T-invariant matrix of the net. It means that if the firing count for transition set $\{ t_i, | 1 \leq i \leq 21 \}$ is equal to one, the final marking of the system is equal to the initial marking of the system. Figure 20 is the P-invariant matrix of the net, and Figure 21 is all possible markings of the net. It indicates that the total marking in the net, when weighted by the P-invariant vector, is constant. This means that the net has an up-bound number of tokens in all the places. Thus the net is bound.

Liveness

Liveness indicates that the system is free of deadlocks. Liveness analysis identifies the places in the net that cause the deadlocks. Once identified, the system can either be redesigned or use some kind of human interference to resolve the deadlock problem. The later is more appropriate in the office environment. The workflow model for the Extension office is shown to be live. Thus, it is deadlock-free.

Reversibility

Table 3. T-Invariant matrix shows that if the firing count for transition set $\{ t_i, | 1 \leq i \leq 21 \}$ is equal to one, the net will go back to its initial marking M_0 . Various simulation results shown that the net is always possible to return to its initial marking M_0 . It thus satisfies the property of reversibility, which means that the system can reinitialize itself, and it has the ability to cover from errors or failures.

Persistence

The general net analysis under PESIM is free of conflict situations. A conflict situation is such that a place in the net can be the input of more than one transitions, for example, the sharing of resources. Figure 19 is an example of conflict situation. It is also called a free choice net. A feedback loop out of a place, P_i , is also a situation where conflict can occur due to the choice of the loop. The enrollment subnet of the workflow does have such a situation. Since the analysis of the workflow only considers the primary processes of the loop as mentioned before, this feedback choice is not to be considered during the net analysis. Thus, the net does not have a conflict situation under such a circumstance. In other words, the net is persistence.

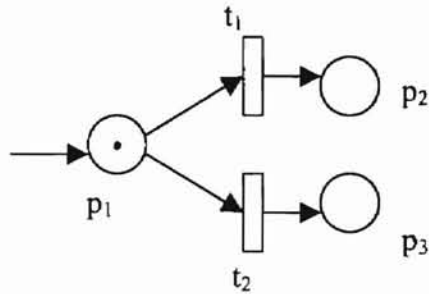
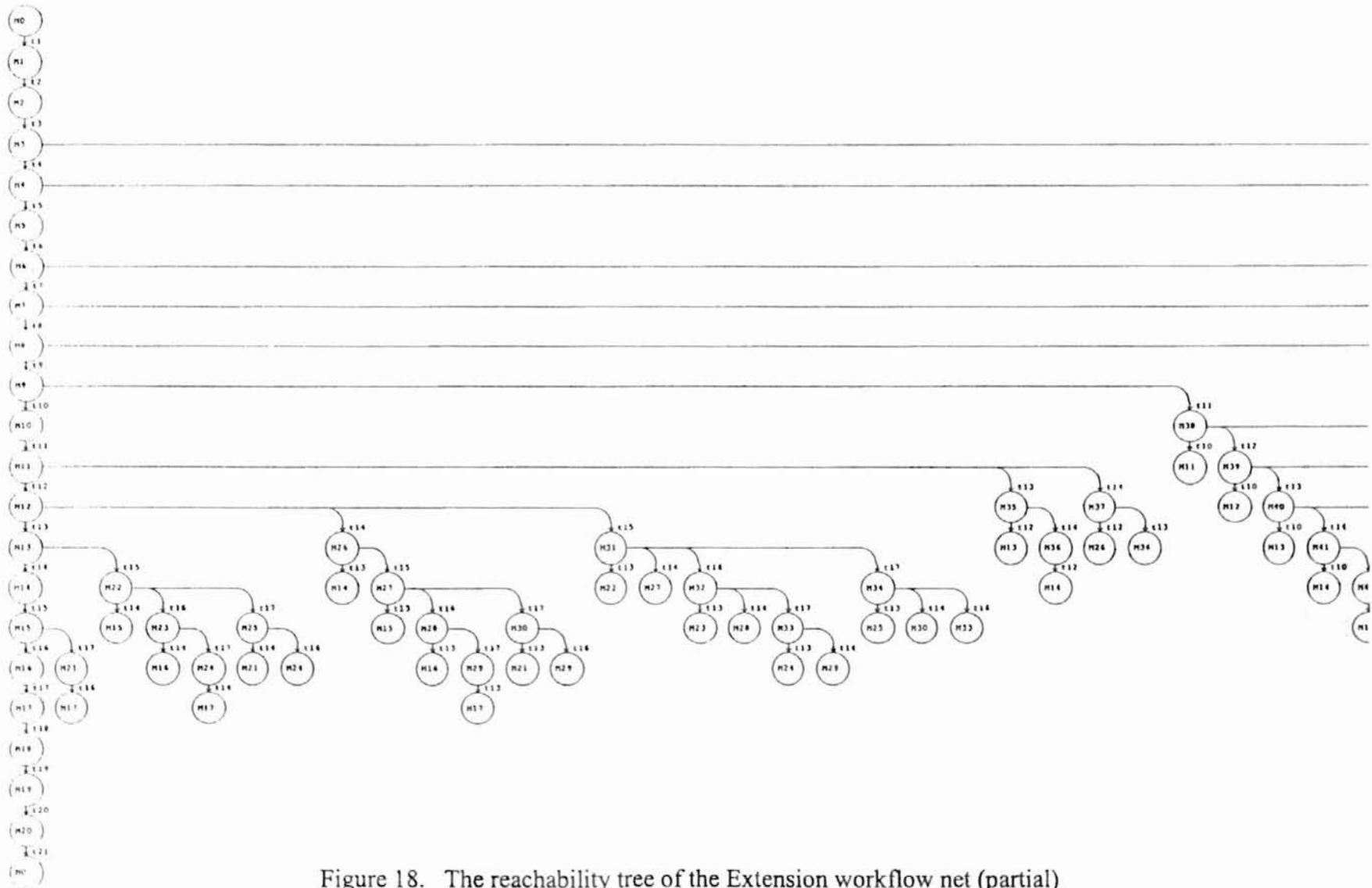


Figure 19. An example of conflict situation. It is also called a free-choice net. A token in p_1 enables t_1 and t_2 . No other tokens in the net influence this decision; therefore, the conflict can be resolved by freely choose which transition gets enabled.

Conservative

A conservative net is one in which the total number of tokens in the system is constant at any given moment. Since the number of tokens in the Extension office workflow changes with the firing of various transitions, it is not a conservative net as indicated by the computation in figure 14. The variation of the number of tokens in the workflow is due to the parallelism in the system.



Invariant	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 3. The T-Invariant of the Extension workflow net

Invariant	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	
1	1	1	1		1		1																			1	1	1	
2	1	1	1	1		1					1				1												1	1	1
3	1	1	1	1		1		1				1					1			1							1	1	1
4	1	1	1	1		1			1				1				1			1							1	1	1
5	1	1	1	1		1				1				1			1			1							1	1	1
6	1	1	1	1		1		1				1						1			1						1	1	1
7	1	1	1	1		1			1				1					1			1						1	1	1
8	1	1	1	1		1				1				1				1			1						1	1	1
9	1	1	1	1		1		1				1			1				1			1			1		1	1	1
10	1	1	1	1		1			1				1			1				1			1		1		1	1	1
11	1	1	1	1		1				1				1			1			1			1		1		1	1	1
12	1	1	1	1		1		1				1				1				1			1		1		1	1	1
13	1	1	1	1		1			1				1				1			1			1		1		1	1	1
14	1	1	1	1		1				1				1			1			1			1		1		1	1	1

Table 4. The P-Invariant of the Extension workflow net

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19	p20	p21	p22	
M0	1																						
M1		1																					
M2			1																				
M3				1	1																		
M4					1	1																	
M5						1	1																
M6							1	1	1	1	1												
M7							1		1	1	1	1											
M8							1			1	1	1	1										
M9							1				1	1	1	1									
M10												1	1	1	1								
M11															1	1	1	1					
M12															1		1	1	1				
M13															1			1	1	1			
M14															1				1	1	1		
M15															1					1	1	1	
M16															1					1	1		
M17															1					1	1		
M18																							
M19																							
M20																							
M21															1					1	1	1	
M22															1			1		1		1	
M23															1			1		1			
M24															1			1		1			
M25															1			1		1			1
M26															1		1		1		1		
M27															1		1				1	1	
M28															1		1				1		
M29															1		1				1		
M30															1		1				1	1	
M31															1		1	1					1
M32															1		1	1					
M33															1		1	1					
M34															1		1	1					1
M35															1	1		1			1		
M36															1	1				1	1		
M37															1	1	1					1	
M38							1				1						1	1					
M39							1				1						1	1	1				
M40							1				1							1	1	1			

Table 5. Partial markings of the Extension workflow net
(see Appendix D for all markings of the system)

CHAPTER V

OTHER ISSUES FOR OFFICE AUTOMATION

The phase I of the system was installed and tested successfully in spring 1998. During the development process, additional features were added: selected printing of nametags, batch printing of invoices, etc. to make it more user friendly. Upon reviewing the case, several parameters appear to be more crucial than the others for the development of office automation.

During the case study, several issues arose due to the implementation of an automation system in the office. Although these issues are not directly related to the success of the new automation system, that could to some extent become a potential major concern in the future, and thus determine the success or failure of the automation system in a long run.

Attitudes and Expectations

Our case study verified Ariano and Dagnino's (1996) claim that the attitude of the staff as well as the management of the office to be automated is important with regard to the success of office automation. In the extension office case, both management and staff were open to new technology, and are supportive of its use in the work environment.

They have been enthusiastic about office automation, and made substantial commitments to the project in terms of hardware and software acquisitions. The system design and selection went smoothly. Budget was approved and allocated without too much bureaucratic paperwork.

Kowal (1992) pointed out that success in developing new systems or software products is determined by the customer. The customer evaluates new systems and software projects by how well they meet his or her expectations. Unrealistic expectations from users would cause user dissatisfaction regardless how well the system performed. Overly higher promises also could lead to later dissatisfaction from users. For example, some staff in the office thought that the automation means that every thing s/he did on the paper in the past would be automated. She/He was surprised when told that in exceptional cases, she/he will have to type the bills on the typewriter. It is important to tell the user what the system can do and what it cannot do to avoid unnecessary frustrations.

User-Driven Approach

A user-driven approach is a design approach that includes the user in the overall system design and analysis. (Tapscott, 1982). In other words, the system leaves some room for the user to make decisions instead of making all the decisions for the users. The case study showed that the more users were involved into the system, the better user satisfaction. Initially, the system was designed to automate almost 99% of the tasks, which appeared to be wonderful from a technical point of view. During the testing period, it was found that the users felt no control over what was happening, the system left little

room for user inputs. It became obvious when exceptional cases occurred during the operational phase of the automated system. It was found that if given enough opportunity for user input, many exceptional cases could be handled more satisfactorily. In the Extension office these exceptional cases were initially headaches for the designers.

The users generally feel more in control when the system leaves some room for the user inputs under certain situations. Since the policies and the rules may change, or the procedures may not be the same at all times, it is necessary to allow office automation software to be flexible in order to accommodate a changing environment. Depending on the user educational or experience level, a balance should to be found between the amount of user inputs and easiness for a new user to learn the system.

Office automation is a continuing process, not a project (Epstein 1991). The case study showed that if we view the automation as a project, it would soon lose its usefulness. Since the environment of the Extension office changes with the changes of the university in terms of the course offered, the offering media, the enrollment policies, etc., the office automation software should be flexible and expandable. Over the years, additional modules will be added, and some modules will become obsolete. Thus, the budget for the system should be a continuous one instead of one-time investment.

Office Automation and Productivity

Chorafas (1982) called office automation systems “productivity engines”. This means getting better quality work done per hour. The case study showed that the implementation of the automation system increased the overall productivity of the office by about 30%, varying from 70% for certain functions and 20% for the others, depend on how many manual tasks were involved before the implementation of the system. The more manual tasks before, the higher the productivity increased after the system implementation. The accuracy of the information and its timeliness ensured the smoothness and effectiveness of daily operations in the office.

One of the managerial functions in the case study was to assess the workload of each staff member. Management needs this information to determine the amount of the work each course/program requires for future reference, and to assess the workload of each staff member. The design and programming part of the function was straightforward, the real issue was the possible effect on the staff member. If the workers feel that being monitored by a computer on hourly basis is an unfair procedure, resentment to automated system could arise. As a result the exaggerated complaining of the minor system problems could lead to the shut down of certain functions.

Office Automation as Hindrance

Lisa Moran (1982) pointed out that office automation could increase the productivity in the office, and at the same time become a source of hindrances. One of the

hindrances due to the implementation of the automation system was the “system down time.” When that happened, the whole office functions had to be shut down due to lack of required information for processing. The other hindrance was the continuous investment on PC/MAC updates. Certain budget must be allocated yearly just to upgrade the computers. When the funds are not available, the system could be very slow and ineffective.

The case study showed that training time is very essential for the proper use of the system. When an integrated system was implemented, everyone in the office must have training to understand the system better and to use it effectively. Due to lack of formal training time, many operational errors occurred causing chain reaction in subsequent functions, giving an appearance that the system was not functioning properly. The problem was alleviated after the source of errors was identified, and several training sessions for the functional groups were arranged.

CHAPTER VI

SUMMARY AND FUTURE WORK

Summary

Office Automation is a complex task that requires a good understanding of both the office environment as an "Open System" (Hewitt 1986) and the office process as "Due Process" - excessive iterations take place before the final decision is made (Gerson and Star, 1986). As the office automation system evolved from local computer, workgroup computing, and workflow computing towards cooperative, intelligent workflow computing, Petri net is a very useful tool for office automation design, modeling, and analysis. The Petri net based workflow system automates and streamlines the office operation by examining its day-to-day office processes and looking to office process reengineering to provide solutions. Though the Petri nets were originally used for industrial control and computer operating systems design, it has found many applications in the business/office as well. It is an expanding research area with new developments in both theory and application each year. Our case study indicates that using Petri net based workflow design, modeling, analysis, and implementing of an office automation system is viable. Its benefits include: ease of use by a non-technical person, a system view for the technical person, formal semantics, and strong analytical tools to prove properties.

Further Research Topics

Because of the complexity of the office task, it is naturally semi-structured. Many procedures do not have clear-cut beginnings and endings. There are many overlaps among the functional units. The Petri net model deals with this overlapping by means of interaction, or a local loop. A local loop represents several iterations within a functional unit. However, it could not represent the overlap among the functional units. Such a net would be very large and complicated, and the reachability tree could grow exponentially. More researches should be done in this area.

SELECTED BIBIOGRAPHY

- Adam, N., Atlure, V., and Huang, W. K. (1998). Modeling and Analysis of Workflows Using Petri Nets. Journal of Intelligent Information Systems, Vol. 10, pp.131-158.
- Ang, J., and Conrath, D. W. (1996). An Office Analysis Methodology Using Petri Nets and Playscripts. IEEE Transactions on System, Man, and Cybernetics – Part A: Systems and Humans. Vol. 25, No. 5, pp. 572-582.
- Ang, J., Vanneste, C., and Lu, G. (1998). End User Dialogue Context Management of Office automation System. Behaviour & Information Technology, Vol. 17, No. 3. pp. 145-151.
- Ariano, M., and Dagnino, A. (1996). An Intelligent Order Entry and Dynamic Bill of Materials System for Manufacturing Customized Furniture. Computers Elect. Engng. Vol 22, No. 1, pp. 45-60.
- Atlure, Vijayalakshmi and Huang, Wei-Kuang (1996). An Extended Petri Net Model for Supporting Workflows in a Multilevel Secure Environment. In Proc. Of the 10th IPIF WG 11.3 Working Conference on Database Security, July 1996.
- Benjamin, R. I. (1982). Information Technology in the 1990s: A Long Range Planning Scenario. Management Information Systems. Q. 6, No. 2, pp. 11-31.
- Bracchi, Giampio, and Pernici, Barbara (1984). The Design Requirements of Office Systems. ACM Transactions on Office Information Systems, Vol. 2, No. 2, pp. 151-170.
- Casati, F., Ceri, S., Pernici, B., and Pozzi, G. (1998). Workflow Evolution. Data & Knowledge Engineering. Vol. 24, No. 3, pp. 211-238, North-Holland, Elsevier Science Publishers.
- Chen, K. Y., and Lu, S. S. (1997). A Petri-Net and Entity-Relationship Diagram Based Object-Oriented Design Method for Manufacturing Systems Control. International Journal of Computer Integrated Manufacturing, Vol. 10, No. 1-4, pp. 17-28.
- Chorafas, Dimitris N. (1982). Office Automation: The Productivity Challenge. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

- Connell, J. J. (1978). The Automated Office: The Challenge. Datamation Vol. 24, No. 11, pp. 162-168.
- Croft, W. Bruce, and Lefkowitz, Lawrence S. (1984). Task Support in an Office System. ACM Transactions on Office Information Systems, Vol. 2, No. 3, pp. 197-212.
- Desrochers, Alan A. and Al-Jaar, Tobert Y. (1995). Applications of Petri Nets in Manufacturing Systems – Modeling, Control, and Performance Analysis. IEEE Control systems Society Press, New York, NY.
- Dicesare, F., Harhalakis, G., Proth, J. M., Silva, M., Vernadat, F. B. (1993). Practice of Petri Nets in Manufacturing, Chapman & Hall, London.
- Drucker, P. (1967). The Effective Executive. Harper & Row.
- Dumais, S., Kraut, R., and Koch, S. (1988). Computers' Impact on Productivity and Work Life. ACM Conference on Office Information Systems: March 23-25, 1988. Palo Alto, California.
- Ellis, C. and Nutt, G. (1993). Modeling and Enactment of Workflow System. Applications and Theory of Petri Nets, M. A. Marsan, Ed., Lecture Notes in Computer Sciences 691. Berlin: Springer-Verlag, 1993, pp.1-16.
- Ellis, C., Gibbs, S. J., and Rein, G. L. (1991). Groupware: Some issues and experiences. Comm. ACM, Vol. 34, No. 1, pp. 38-58.
- Epstein, Susan B. (1991). Managing Technology: Administrators of Automated Systems – A Job Description. Library Journal. March 15, 1991.
- Georgakopoulos, Dimitrios and Hornick, Mark (1995). An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases, pp. 119-153.
- Georgakopoulos, Dimitrios, Hornick, Mark F., and Manola, Frank (1996). Customizing Transaction Models and Mechanisms in a Programmable Environment Supporting Reliable Workflow Automation. IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 4, pp.: 630-649.
- Gerson, Elihu M. and Star, Susan L. (1986). Analyzing Due Process in the Workplace, ACM Transactions on Office Information Systems, Vol. 4, No. 3, pp. 257-270.
- Goldfield, R.J. (1985). Aiming OA towards the Top. Modern Office Technology, Vol. 30, No. 2, pp. 55ff.

- Gremillio, L. L. and Pyburn, P. J. (1985). Justifying Decision Support and Office Automation Systems. Journal of Management Information Systems, Vol. 2, No. 1, pp 5-17.
- Gruhn, Volker (1995). Business Process Modeling and Workflow Management. International Journal of Cooperative Information Systems. Vol. 4, No. 2 & 3, pp. 145-164, World Scientific Publishing Company, United Kingdom.
- Harris, S. E., and Brightman, H. J. (1985). Design Implications of a Task-Driven Approach to Unstructured Cognitive Tasks in Office Work. ACM Transactions on Office Information Systems, Vol. 3, No. 3, pp. 292-306.
- Henderson-Sellers, B., and Constantine, L. L., (1990). Object-oriented Development and Functional Decomposition. Journal of Object-Oriented Programming, Vol. 33, No. 9, pp. 142-169.
- Hewitt, Carl (1986), Office Are Open Systems. ACM Transactions on Office Information Systems, Vol. 4, No. 3, pp. 271-287.
- Hirschheim, R. A. (1986). Understanding the Office: A Social-Analytic Perspective. ACM Transactions on Office Information Systems, Vol. 4, No. 4, pp. 331-334.
- Hsu, M. C. and Kleissner, C. (1996). Object Flow towards A Process Management Infrastructue. Distributed and Parallel Databases, Vol. 4, No. 2, April 1996, pp.169-194.
- Ishiguro, Y., Tarumi, H., Asakura, T. Kida, K., Kusui, D., and Yoshifu, K. (1997). Office Work Support with Software Agents. NEC Research & Development. Vol. 38, No. 2, pp.253-261.
- Jarrett, D. (1984). The Electronic Office, Gower, Aldershot, England.
- Jensen, K. (1996). The Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, EATCS Monographs on Theoretical Computer Science, Berlin, Springer-Verlag.
- Judge, D. W., Odgers, B. R., Shepherdson, J. W., Cui, Z. (1998). Agent-Enhanced Workflow. BT Technology J. Vol. 16, No. 3.
- Korth, Henry F. and Silberschatz, Abraham (1997). Database System Concepts. Computer Science Series, McGraw-Hill, Inc. New York.
- Koulopoulos, Thomas M. (1995). The Workflow Imperative: Building Real Word Business Solutions. Van Nostrand Reinhold, International Thomason Publishing Inc. New York.

- Kowal, James A. (1992). Behavior Models: Specifying User's Expectations. Prentice Hall, Englewood cliffs, New Jersey.
- Langer, Arthur M (1997). The Art of Analysis. Springer-Verlag New York, Inc.
- Law, David, and Gorla, Narasimhaiah (1996). Exploring Factors Underlying Effective Office Information systems. Information & Management. Vol. 31, pp. 25-35.
- Mahling, Dirk E., Craven, N., and Croft, W. B. (1995). From Office Automation to Intelligent Workflow Systems. IEEE Expert – Intelligent Systems & Their Applications, Vol. 10, No. 3, pp. 41-47.
- Martin, J., and McClure, C. (1988). Structured Techniques: The Basis for CASE. Prentice-Hall, Englewood Cliffs, New Jersey.
- McLeod, Raymond and Jones, Jack W. (1987). A Framework for Office Automation. MIS Quarterly, Vol 11, pp. 87-104.
- Moran, Lisa N. (1982). Office Automation: Debunking the Myths Concerning its Impact on the Individual and the Organization, a Case Study. Ph.D. Thesis, Northwestern University.
- Murata, Tadao (1989). Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE, Vol. 77, No. 4, pp. 541-580, April 1989.
- Nierstrasz, C., Gibbs, S., and Tsichitris, D. (1992). Component-Oriented Software Development, Comm. ACM, Vol. 35, No. 9, pp. 160-165.
- Nutt, Gary J. (1992). Centralized and Distributed Operating Systems. Prentice Hall, Englewood Cliffs, New Jersey.
- Obrien, P. D. and Wiegand, M. E. (1996). Agents of Change in Business Process Management. BT Technology Journal, Vol. 14, No. 4, pp. 33-140.
- Otway, Harry J. and Peltu, Malcolm (1983). New Office Technology: Human and Organizational Aspects. ECSC, EEC, EAEC, Brussels and Luxembourg.
- Paddock, Charles E. (1984). Office Automation Projects and Their Impact on Organization, Planning, and Control. ACM Transactions on Office Information System, Vol. 2, No. 4, pp. 289-302.
- Panko, Raymond R. (1984). 38 Offices: Analyzing Needs in Individual Offices. ACM Transactions on Office Information Systems, Vol. 2, No. 3, pp. 226-234.
- Peterson, J. L. (1981). Petri Net Theory and the Modeling of Systems. Prentice-Hall, Englewood Cliffs, New Jersey.

- Reichert, M., and Dadam, P. (1998). ADEPT_{flex} – Supporting Dynamic Changes of Workflows Without Losing Control. Journal of Intelligent Information Systems, Vol. 10, pp. 93-129.
- Reisig, W. (1986). Petri Nets in Software Engineering, Petri Nets: Applications and Relationships to Other Models of Concurrency, Part II, Lecture Notes in Computer Science 255, Berlin: Springer-Verlag, pp. 63-96.
- Rockart, J. (1979). Chief Executives Define Their Own Data Needs. Harvard Business Review. Vol. 57, Vol. 2.
- Rozenberg, G. (1986). Behavior of elementary net systems in Petri Nets: Control Models and Their Properties, Part I. Lecture Notes in Computer Science 254. Berlin: Springer-Verlag, pp. 60-94.
- Rusinkiewicz, M., Helal, S. (1998). Workflow Management-System: Introduction. Journal of Intelligent Information Systems. Vol. 10, No. 2, pp. 91-92.
- Sasso, W., Olson, J. and Merten, A. (1986). Procedures for Office Analysis: A Critical Review. Technical Report 119, New York University, Graduate School of Business Administration, CRIS.
- Scheer, A., and Hars, A. (1992). Extending Data Modeling to Cover the Whole Enterprise. Comm. ACM, Vol. 35, No. 9, pp. 166-171.
- Shan, M., Davis, J. W., Du, W., and Chen Q. (1996). Business Process Flow Management and its Application in the Telecommunications Management Network. Hewlett-Packard Journal, October 1996, pp. 70-76.
- Sheth, Amit, and Georgakopoulos, Dimitrios (1996). Report from the NSF Workshop on Workflow and Process Automation in Information Systems. Computer Science Department Technical Report UGA-CS-TR-96-003, University of Georgia, Athens, Georgia, October 1996.
- Sirbu, M. A., Schoichet, S. R., and Kunin, J. S. (1983). Office Analysis: Methodology and Cases. MIT/LCS/ TR-289, Mar. 1983.
- Skacel, M. (1992). Tools for Petri Nets Drawing, Simulation and Analysis Under MS Windows, dipl. Thesis, Department of Computer Science and Engineering, Technical University of Brno, Czechoslovakia.
- Suchman, Lucy A. (1983). Office Procedure as Practical Action: Models of Work and System Design, ACM Transactions on Office Information Systems, Vol. 1, No. 4, pp. 320-328.

- Sutherland, J. (1983). An Office Analysis and Diagnosis Methodology. MIT/LCS/ TR-290, Mar. 1983.
- Sutherland, J., Sirbu, M. (1983). Evaluation of an Office Analysis Technology. Technical Report LSC/TM-239, MIT.
- Tapscott, Don (1982). Office Automation: A User-Driven Method. Plenum Press, New York.
- Van Der Aalst, W. M. P. (1998). The Application of Petri Nets to Workflow Management. Journal of Circuits, Systems, and Computers, Vol. 8, No. 1, pp:21-66.
- Van Der Aalst, W. M. P.(1995). A Class of Petri Net for Modeling and Analyzing Business Processes. Computer Science Report, Eindhoven University of Technology, Eindhoven, 1995.
- Van Der Aalst, W. M. P.(1994). Petting Petri Nets to Work in Industry, Computers in Industry, Vol. 25, No. 1, pp. 45-54.
- Voss, K. (1986). Nets in Office Automation. Petri Nets: Applications and Relationships to Other Models of Concurrency, Lecture Notes in Computer Science 255. Berlin: Springer-Verlag, pp. 234-257.
- Yang, F. C., and Ho, Y. K. (1997). A Cooperative Distributed Problem-Solving Management Framework for Office Automation Systems. Concurrent Engineering – Research and Applications. Vol. 5, No. 1, pp. 77-94.

APPENDIXES

APPENDIX A

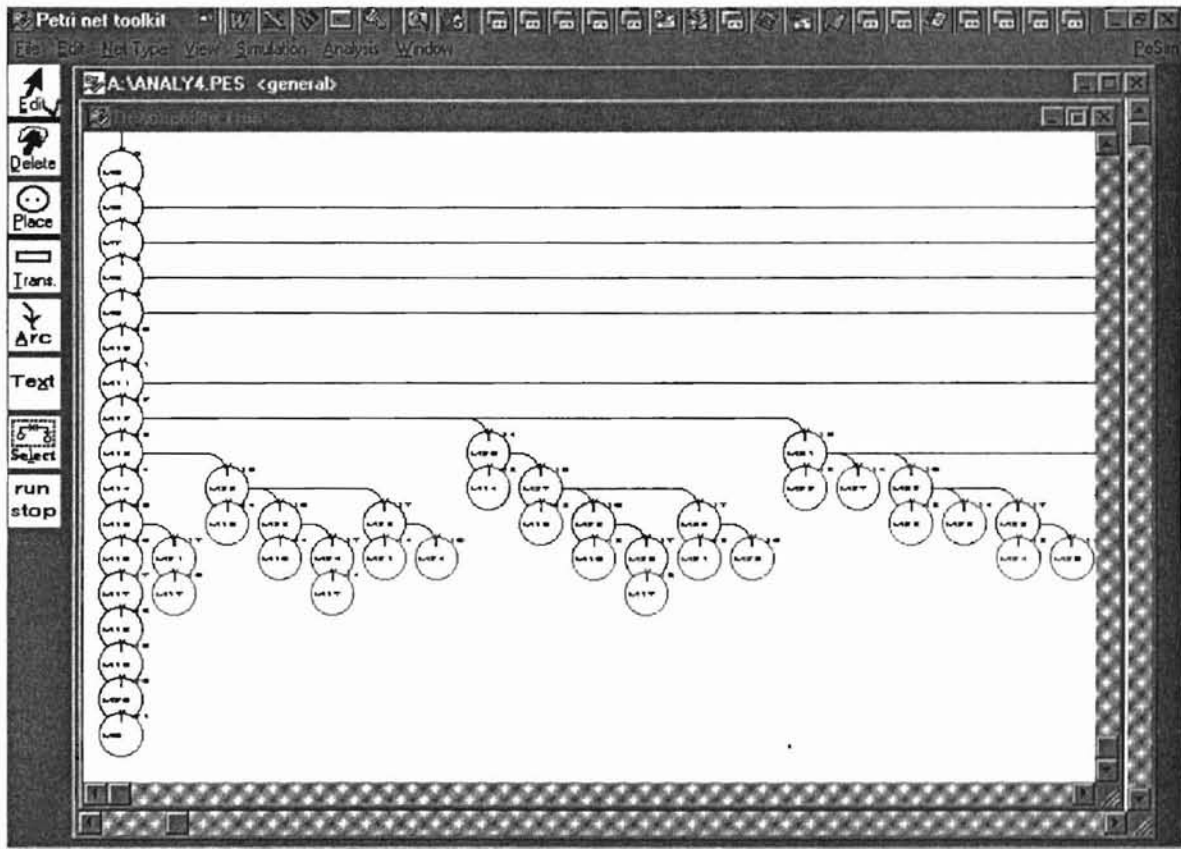
THE EXTENSION WORKFLOW NET IN TEXT FORM OF VIEW

```
Places =
{p1(1),p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12,p13,p14,p15,p16,p17,p18,p19,
p20,p21,p22,p23,p24,p25,p26,p27,p28}
Transitions =
{t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16,t17,t18,t19,t20,
,t21}
p1 -> t1 -> p2;
p2 -> t2 -> p3;
p3 -> t3 -> p4,p5;
p4 -> t4 -> p6;
p5 -> t5 -> p7;
p6 -> t6 -> p8,p9,p10,p11;
p8 -> t7 -> p12;
p9 -> t8 -> p13;
p10 -> t9 -> p14;
p11 -> t10 -> p15;
p12,p13,p14 -> t11 -> p16,p17,p18;
p16 -> t12 -> p19;
p17 -> t13 -> p20;
p18 -> t14 -> p21;
p19 -> t15 -> p22,p23;
p22 -> t16 -> p24;
p23 -> t17 -> p25;
p7,p15,p20,p21,p24,p25 -> t18 -> p26;
p26 -> t19 -> p27;
p27 -> t20 -> p28;
p28 -> t21 -> p1;
```

APPENDIX B

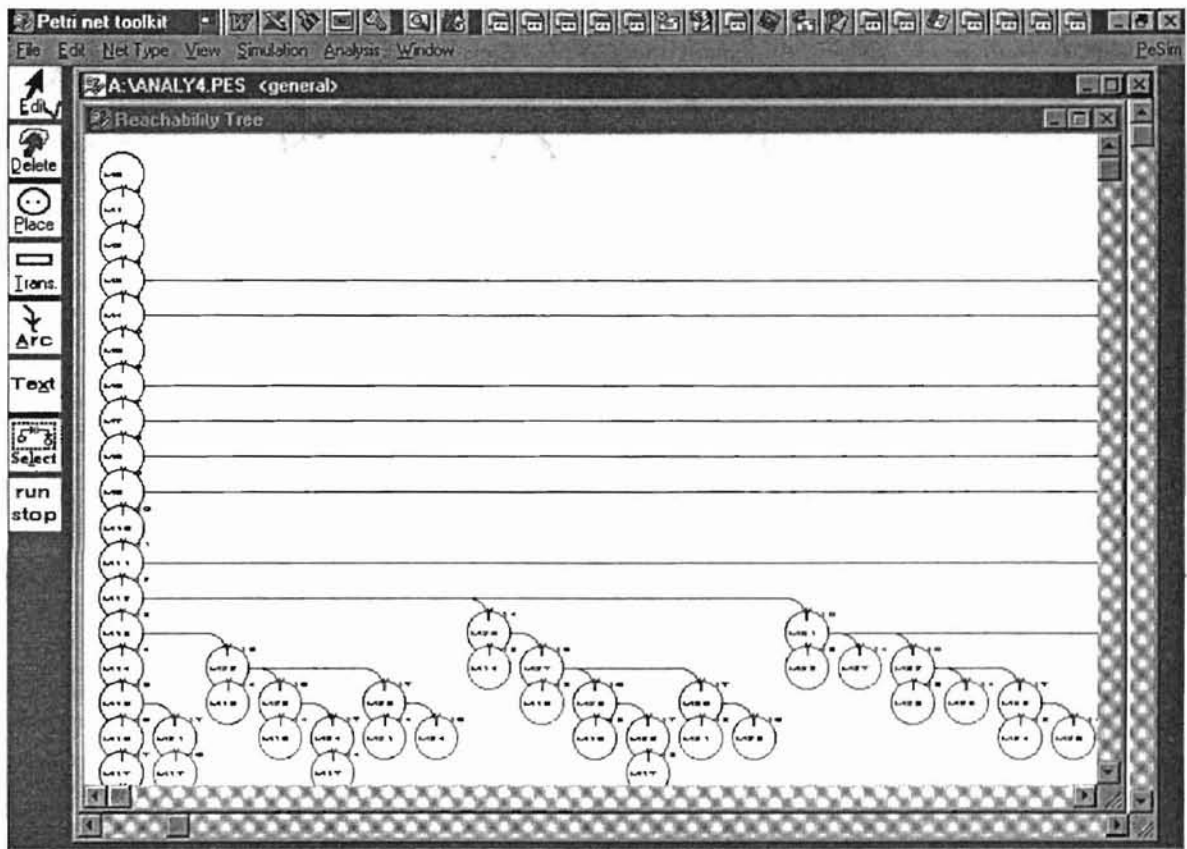
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 1



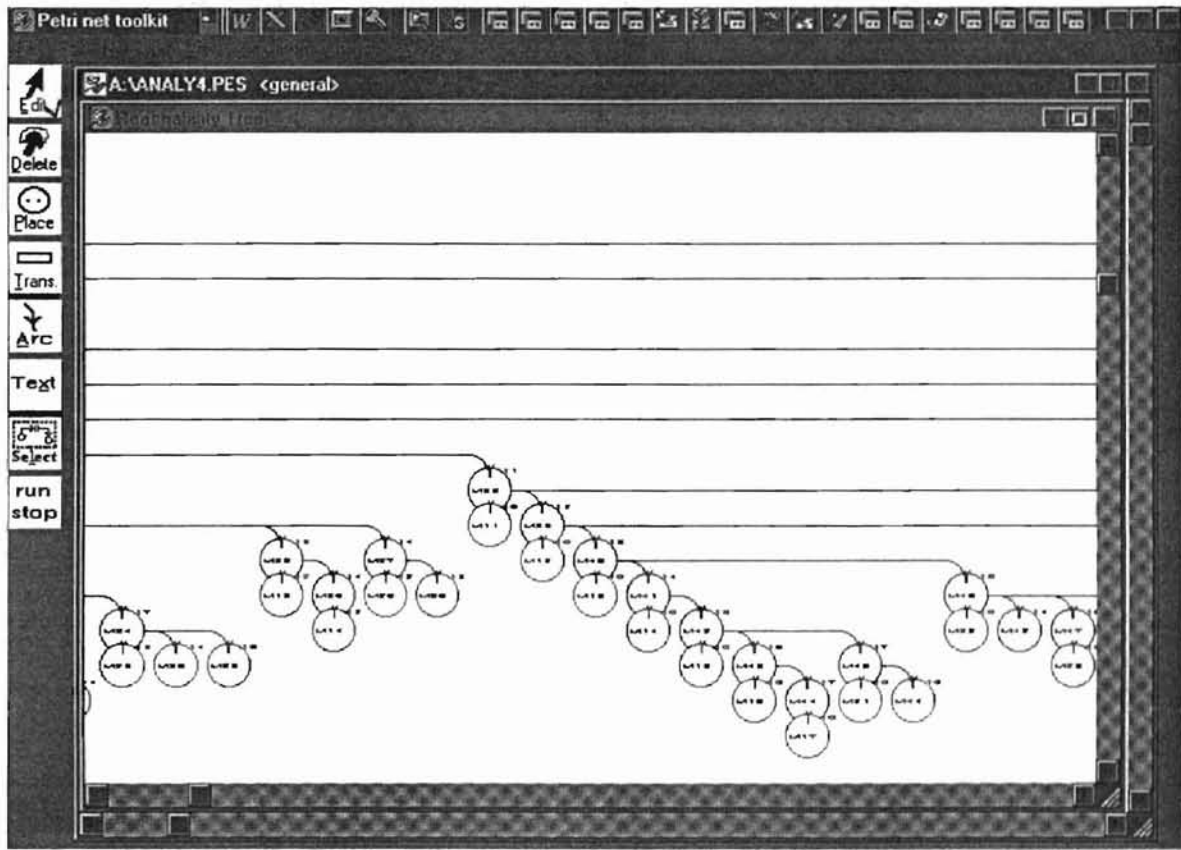
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 2



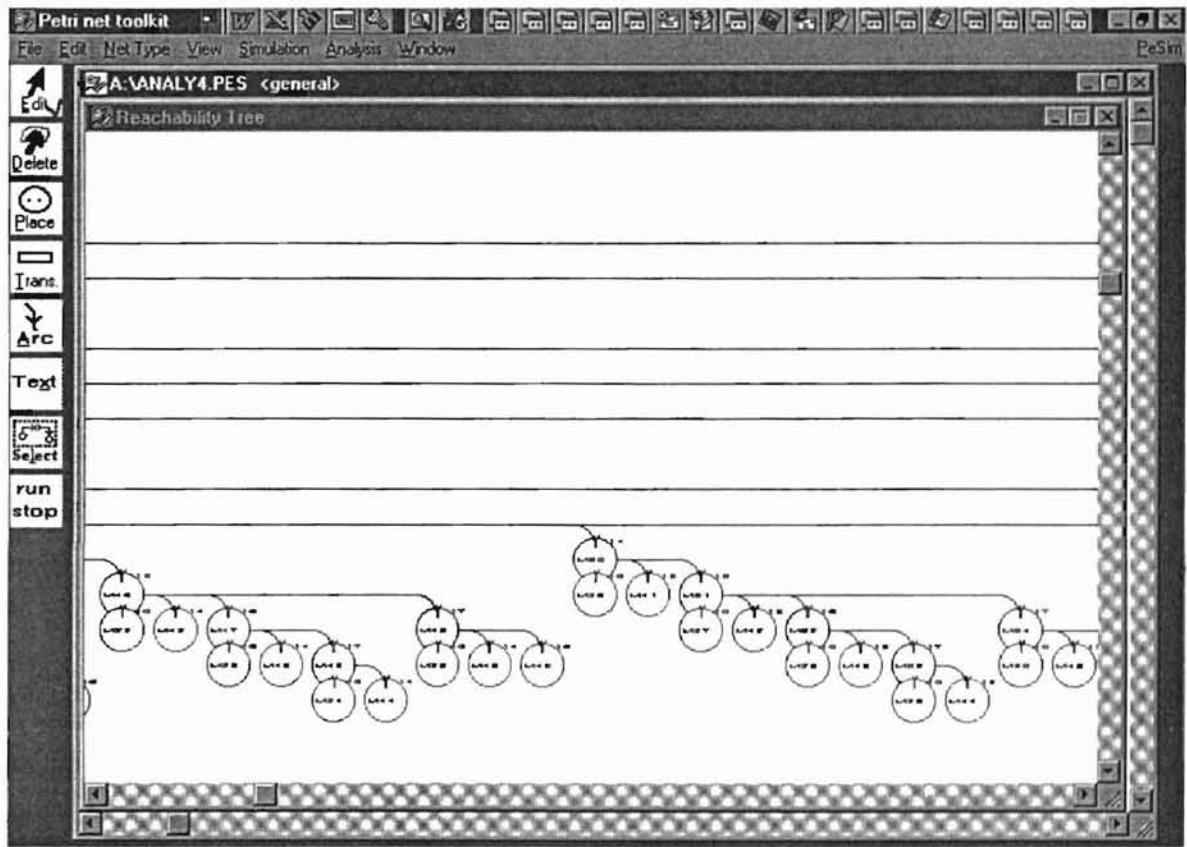
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 3



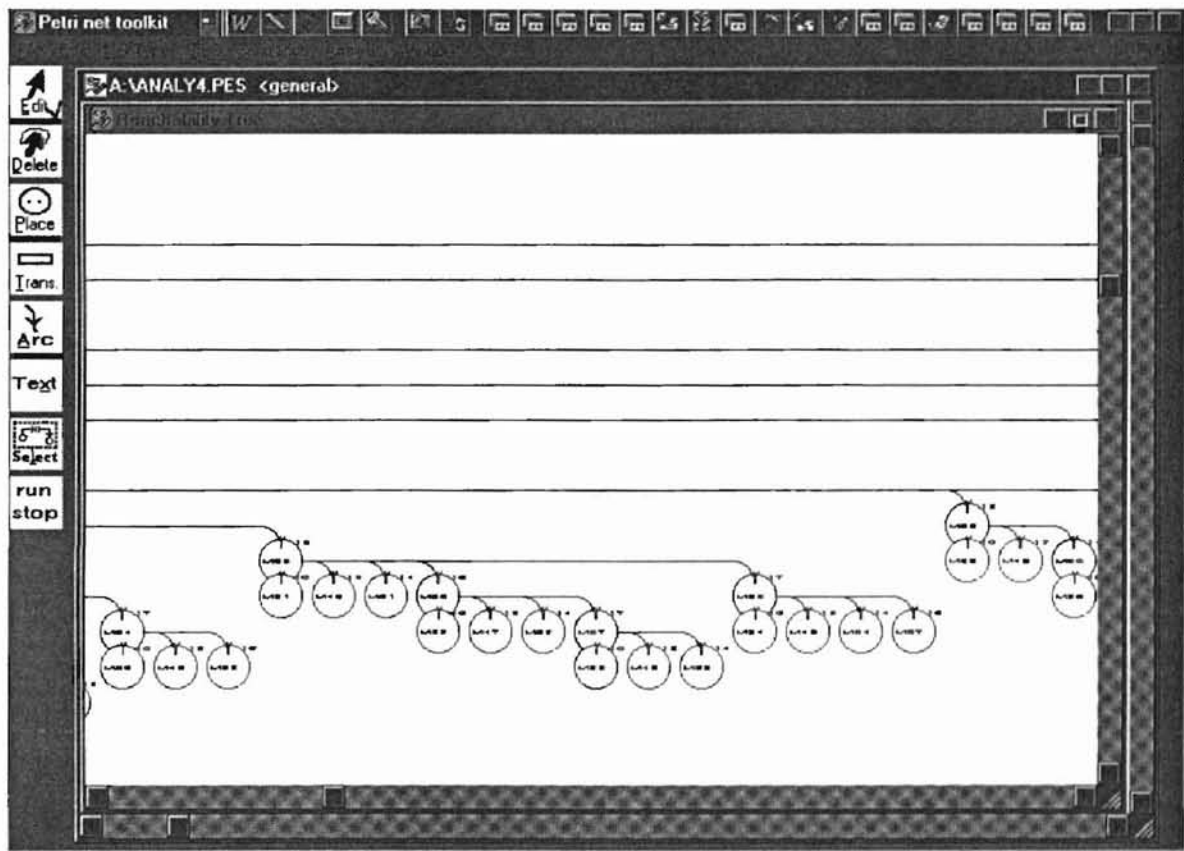
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 4



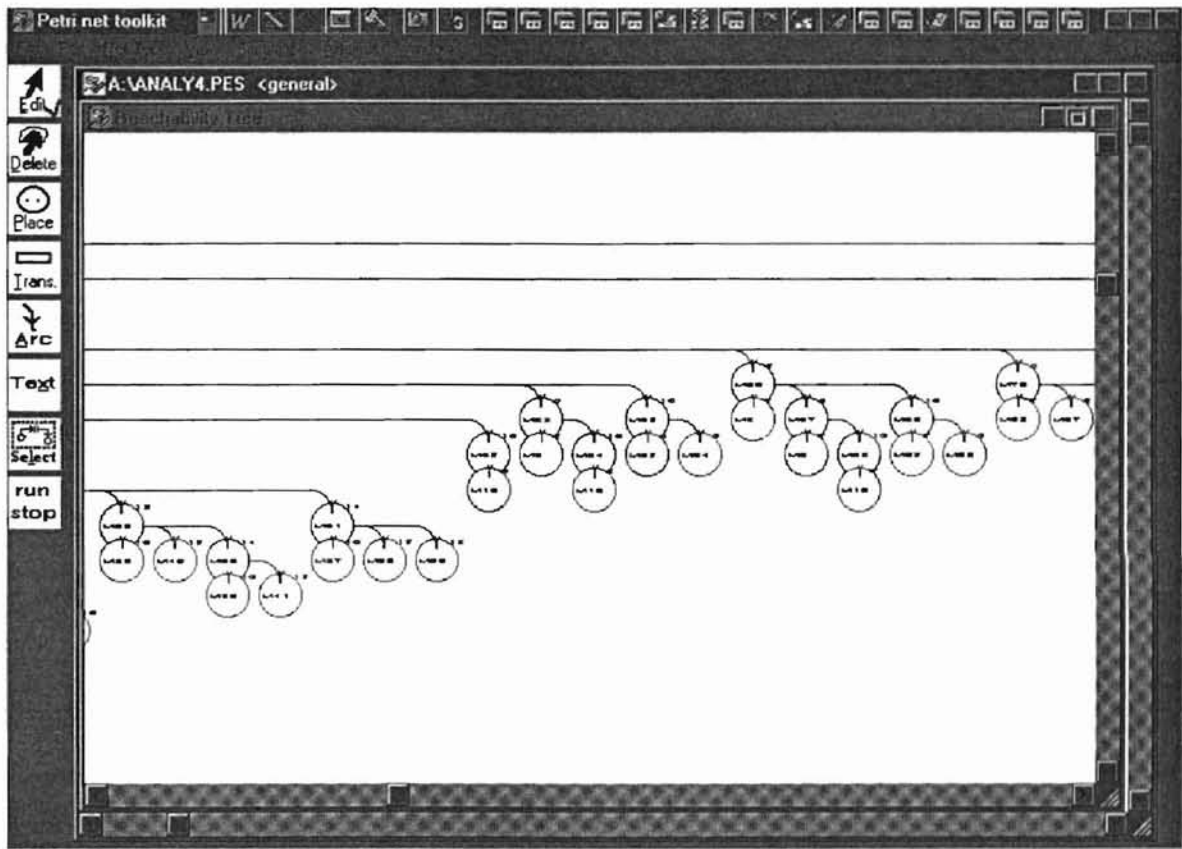
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 5



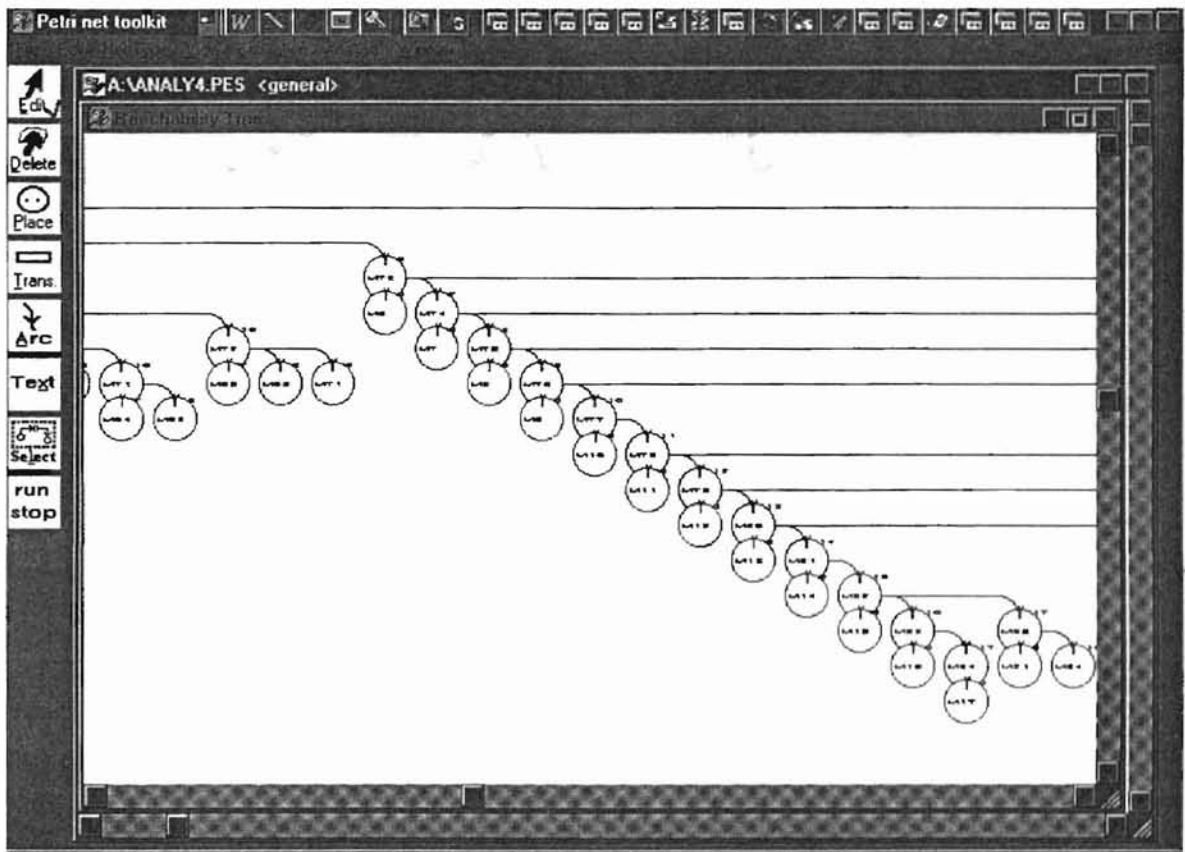
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 6



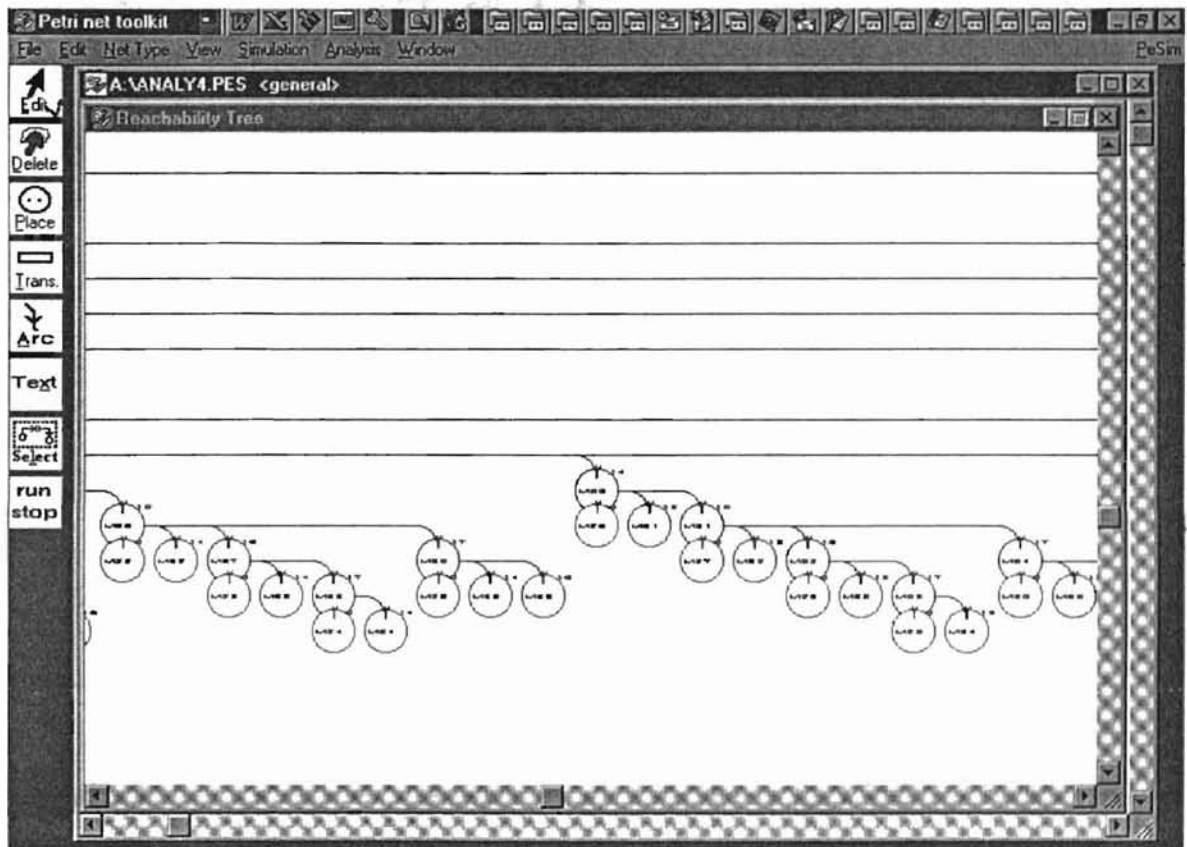
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 7



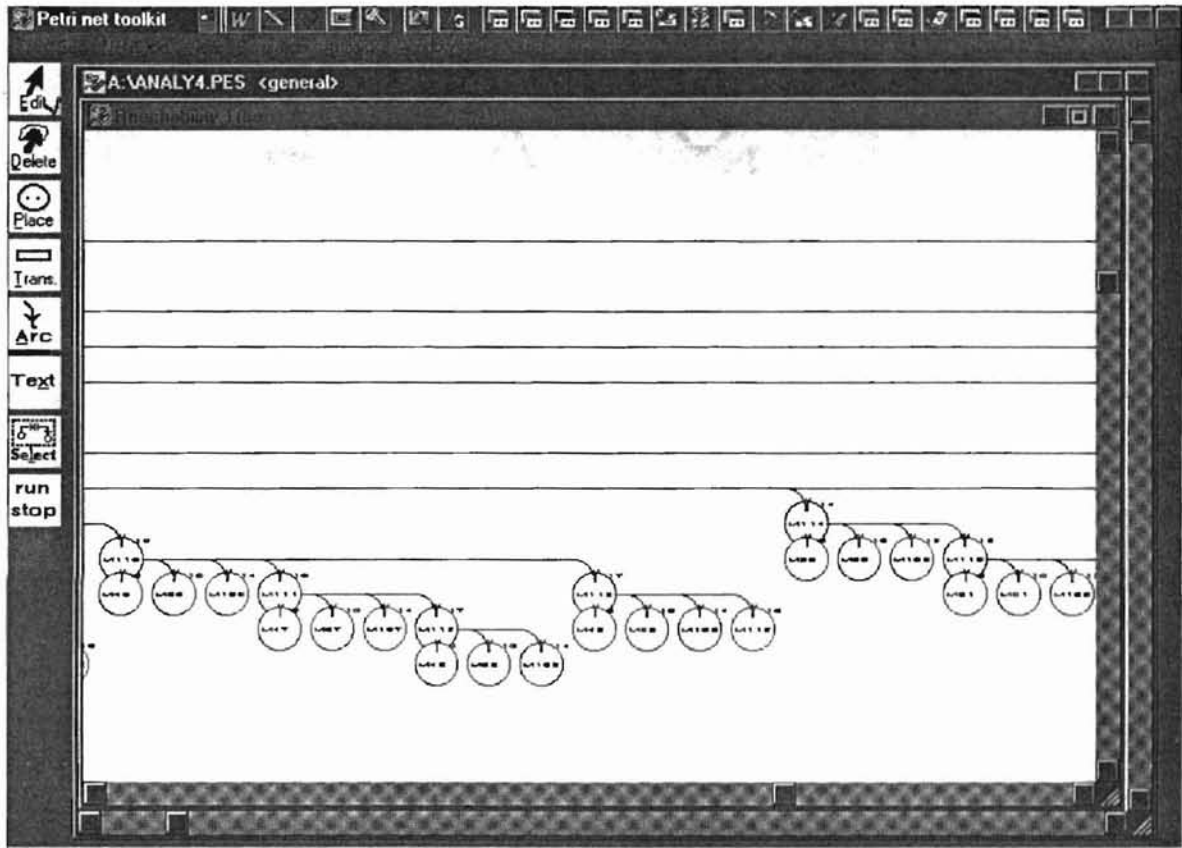
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 8



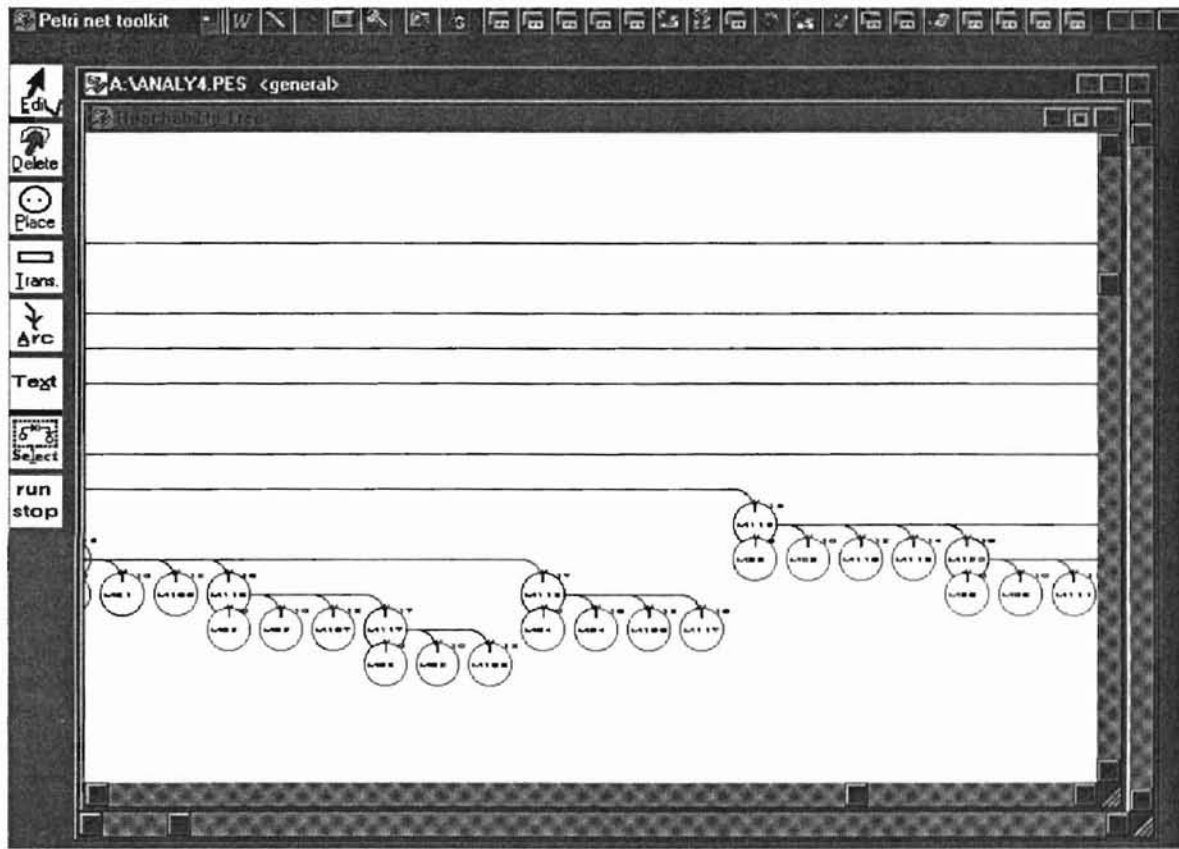
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 11



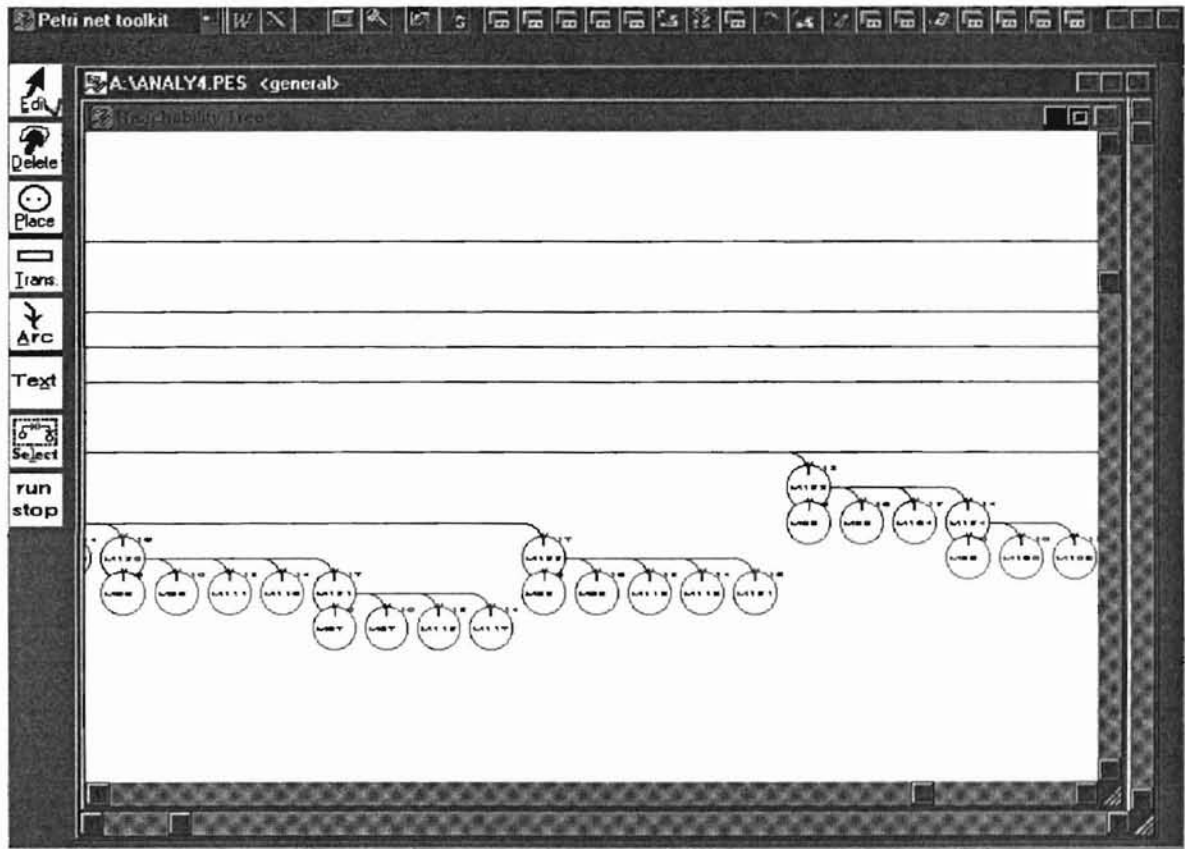
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 12



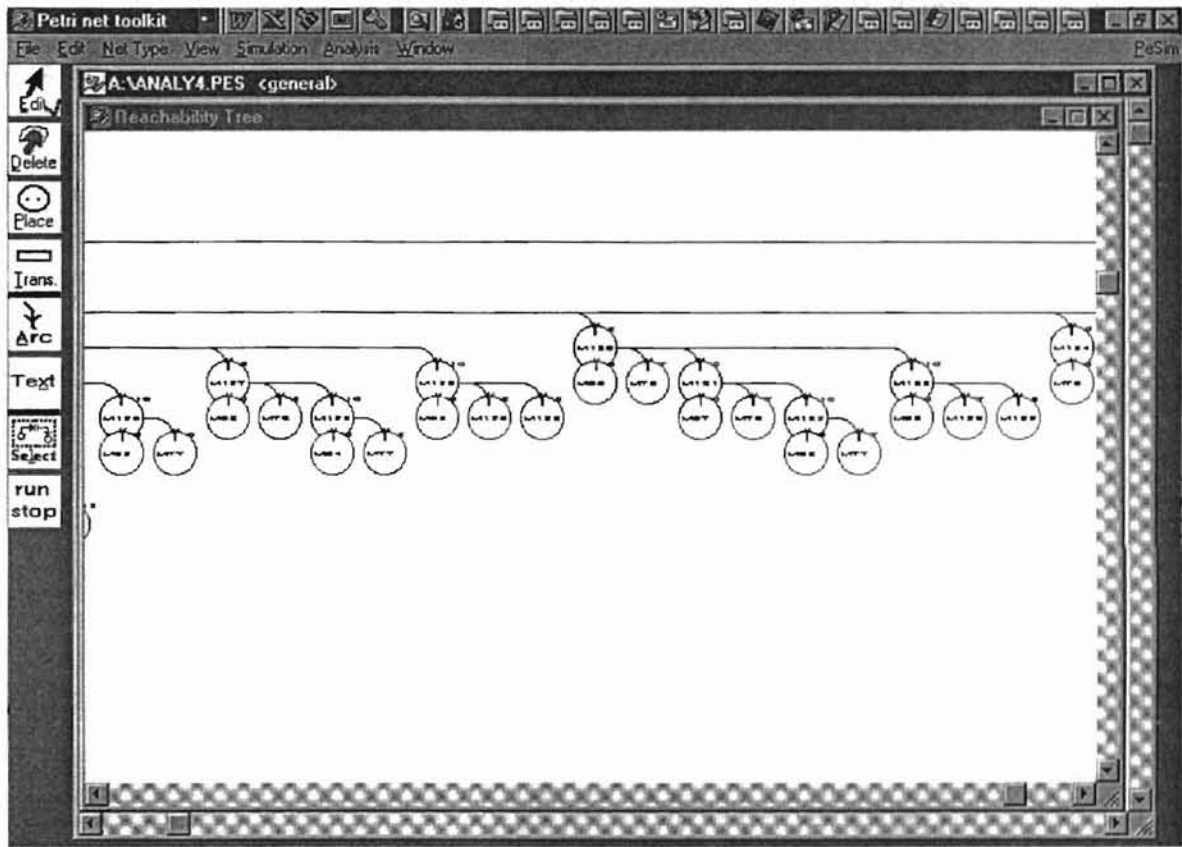
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 13



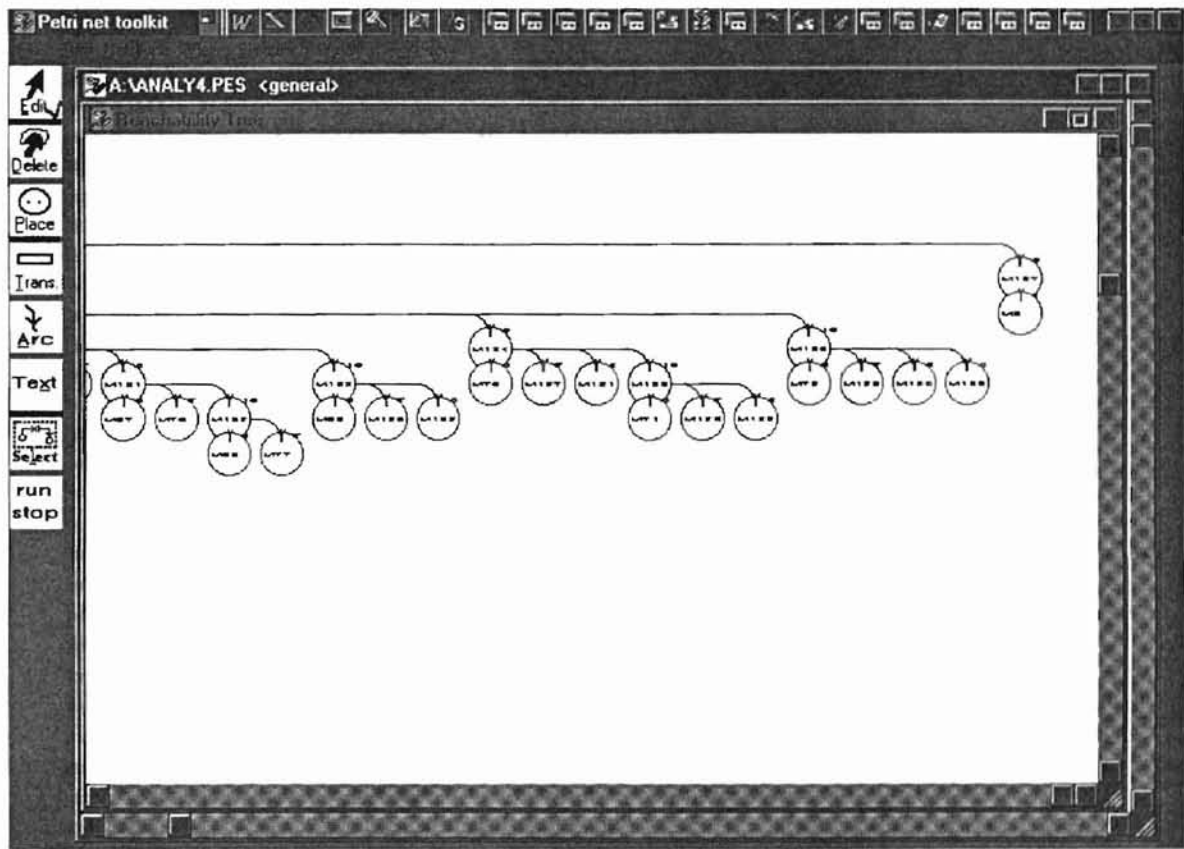
THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 14



THE REACHABILITY TREE IN GRAPHICAL VIEW

Part 15



APPENDIX C

THE REACHABILITY TREE IN TEXT FORM OF VIEW (Part 1)

H0 [t1 > H1			
H1 [t2 > H2			
H2 [t3 > H3			
H3 [t4 > H4	H3 [t5 > H105		
H4 [t5 > H5	H4 [t6 > H73		
H5 [t6 > H6			
H6 [t7 > H7	H6 [t8 > H66	H6 [t9 > H70	H6 [t10 > H72
H7 [t8 > H8	H7 [t9 > H63	H7 [t10 > H65	
H8 [t9 > H9	H8 [t10 > H62		
H9 [t10 > H10	H9 [t11 > H38		
H10 [t11 > H11			
H11 [t12 > H12	H11 [t13 > H35	H11 [t14 > H37	
H12 [t13 > H13	H12 [t14 > H26	H12 [t15 > H31	
H13 [t14 > H14	H13 [t15 > H22		
H14 [t15 > H15			
H15 [t16 > H16	H15 [t17 > H21		
H16 [t17 > H17			
H17 [t18 > H18			
H18 [t19 > H19			
H19 [t20 > H20			
H20 [t21 > H0			
H21 [t16 > H17			
H22 [t14 > H15	H22 [t16 > H23	H22 [t17 > H25	
H23 [t14 > H16	H23 [t17 > H24		
H24 [t14 > H17			
H25 [t14 > H21	H25 [t16 > H24		
H26 [t13 > H14	H26 [t15 > H27		
H27 [t13 > H15	H27 [t16 > H28	H27 [t17 > H30	
H28 [t13 > H16	H28 [t17 > H29		
H29 [t13 > H17			
H30 [t13 > H21	H30 [t16 > H29		
H31 [t13 > H22	H31 [t14 > H27	H31 [t16 > H32	H31 [t17 > H34
H32 [t13 > H23	H32 [t14 > H28	H32 [t17 > H33	
H33 [t13 > H24	H33 [t14 > H29		
H34 [t13 > H25	H34 [t14 > H30	H34 [t16 > H33	
H35 [t12 > H33	H35 [t14 > H36		
H36 [t12 > H34			
H37 [t12 > H26	H37 [t13 > H36		
H38 [t10 > H11	H38 [t12 > H39	H38 [t13 > H59	H38 [t14 > H61
H39 [t10 > H12	H39 [t13 > H40	H39 [t14 > H50	H39 [t15 > H55
H40 [t10 > H13	H40 [t14 > H41	H40 [t15 > H46	
H41 [t10 > H14	H41 [t15 > H42		
H42 [t10 > H15	H42 [t16 > H43	H42 [t17 > H45	
H43 [t10 > H16	H43 [t17 > H44		
H44 [t10 > H17			
H45 [t10 > H21	H45 [t16 > H44		
H46 [t10 > H22	H46 [t14 > H42	H46 [t16 > H47	H46 [t17 > H49
H47 [t10 > H23	H47 [t14 > H43	H47 [t17 > H48	
H48 [t10 > H24	H48 [t14 > H44		
H49 [t10 > H25	H49 [t14 > H45	H49 [t16 > H48	
H50 [t10 > H26	H50 [t13 > H41	H50 [t15 > H51	

THE REACHABILITY TREE IN TEXT FORM OF VIEW (part 2)

K51 [t10 > K27	K51 [t13 > K42	K51 [t16 > K52	K51 [t17 > K54
K52 [t10 > K28	K52 [t13 > K43	K52 [t17 > K53	
K53 [t10 > K29	K53 [t13 > K44		
K54 [t10 > K30	K54 [t13 > K45	K54 [t16 > K53	
K55 [t10 > K31	K55 [t13 > K46	K55 [t14 > K51	K55 [t14 > K56
K56 [t10 > K32	K56 [t13 > K47	K56 [t14 > K52	K56 [t17 > K57
K57 [t10 > K33	K57 [t13 > K48	K57 [t14 > K53	
K58 [t10 > K34	K58 [t13 > K49	K58 [t14 > K54	K58 [t16 > K57
K59 [t10 > K35	K59 [t12 > K40	K59 [t14 > K60	
K60 [t10 > K36	K60 [t12 > K41		
K61 [t10 > K37	K61 [t12 > K50	K61 [t13 > K60	
K62 [t9 > K10			
K63 [t8 > K9	K63 [t10 > K64		
K64 [t8 > K10			
K65 [t8 > K62	K65 [t9 > K64		
K66 [t7 > K8	K66 [t9 > K67	K66 [t10 > K69	
K67 [t7 > K9	K67 [t10 > K68		
K68 [t7 > K10			
K69 [t7 > K62	K69 [t9 > K68		
K70 [t7 > K63	K70 [t8 > K67	K70 [t10 > K71	
K71 [t7 > K64	K71 [t8 > K68		
K72 [t7 > K65	K72 [t8 > K69	K72 [t9 > K71	
K73 [t5 > K6	K73 [t7 > K74	K73 [t8 > K102	K73 [t9 > K104
K74 [t5 > K7	K74 [t8 > K75	K74 [t9 > K101	
K75 [t5 > K8	K75 [t9 > K76		
K76 [t5 > K9	K76 [t11 > K77		
K77 [t5 > K38	K77 [t12 > K78	K77 [t13 > K98	K77 [t14 > K100
K78 [t5 > K39	K78 [t13 > K79	K78 [t14 > K99	K78 [t15 > K94
K79 [t5 > K40	K79 [t14 > K80	K79 [t15 > K85	
K80 [t5 > K41	K80 [t15 > K81		
K81 [t5 > K42	K81 [t16 > K82	K81 [t17 > K84	
K82 [t5 > K43	K82 [t17 > K83		
K83 [t5 > K44			
K84 [t5 > K45	K84 [t16 > K83		
K85 [t5 > K46	K85 [t14 > K81	K85 [t16 > K86	K85 [t17 > K88
K86 [t5 > K47	K86 [t14 > K82	K86 [t17 > K87	
K87 [t5 > K48	K87 [t14 > K83		
K88 [t5 > K49	K88 [t14 > K84	K88 [t16 > K87	
K89 [t5 > K50	K89 [t13 > K80	K89 [t15 > K90	
K90 [t5 > K51	K90 [t13 > K81	K90 [t16 > K91	K90 [t17 > K93
K91 [t5 > K52	K91 [t13 > K82	K91 [t17 > K92	
K92 [t5 > K53	K92 [t13 > K83		
K93 [t5 > K54	K93 [t13 > K84	K93 [t16 > K92	
K94 [t5 > K55	K94 [t13 > K85	K94 [t14 > K90	K94 [t14 > K95
K95 [t5 > K56	K95 [t13 > K86	K95 [t14 > K91	K95 [t17 > K94
K96 [t5 > K57	K96 [t13 > K87	K96 [t14 > K92	
K97 [t5 > K58	K97 [t13 > K88	K97 [t14 > K93	K97 [t14 > K94
K98 [t5 > K59	K98 [t12 > K79	K98 [t14 > K99	
K99 [t5 > K60	K99 [t12 > K80		
K100 [t5 > K61	K100 [t12 > K89	K100 [t13 > K99	
K101 [t5 > K62	K101 [t8 > K76		
K102 [t5 > K66	K102 [t7 > K75	K102 [t9 > K103	
K103 [t5 > K67	K103 [t7 > K76		
K104 [t5 > K70	K104 [t7 > K101	K104 [t8 > K103	
K105 [t4 > K5			

APPENDIX D

ALL MARKINGS OF THE SYSTEM

VITA

Ye-Qiong Zhu

Candidate for the Degree of

Master of Science

Thesis: AN INTEGRATED FRAMEWORK FOR OFFICE AUTOMATION

Major Field: Computer Science

Biographical:

Personal Data: Born in Shanghai, P. R. China, the daughter of Enliang Zhu and Wang Dai-Ming.

Education: Graduated from Shanghai Technology Institute, Shanghai, P. R. China; received Bachelor of engineering in Electrical and Electronics Engineering in July 1983. Received Master of Business Administration degree from Oklahoma State University, Stillwater, Oklahoma, USA. in May 1993. Completed the requirements of the Master of Science degree with a major in Computer Science at Oklahoma State University in July 1999.

Experience: Employed by Oklahoma State University, Arts & Sciences Extension as a Financial Coordinator, Oct 94 to present. Employed by Consortium International Inc., Oklahoma City as a Financial Analyst/accountant, Jan 93 to Sept. 94. Participated in research project at Fluid Power Research Center, OSU, Aug 86 to July 88 as a Visiting Researcher from China. Employed by Shanghai Automobile Research Institute, P. R. China as an Electrical Engineer, Aug 83 to July 86.

Professional Memberships: ACM (Association of Computer Machinery), Beta Gamma sigma (Honor Society for Business Graduates).