

## Copyright Undertaking

This thesis is protected by copyright, with all rights reserved.

**By reading and using the thesis, the reader understands and agrees to the following terms:**

1. The reader will abide by the rules and legal ordinances governing copyright regarding the use of the thesis.
2. The reader will use the thesis for the purpose of research or private study only and not for distribution or further reproduction or any other purpose.
3. The reader agrees to indemnify and hold the University harmless from and against any loss, damage, cost, liability or expenses arising from copyright infringement or unauthorized usage.

If you have reasons to believe that any materials in this thesis are deemed not suitable to be distributed in this form, or a copyright owner having difficulty with the material being included in our database, please contact [lbsys@polyu.edu.hk](mailto:lbsys@polyu.edu.hk) providing details. The Library will look into your claim and consider taking remedial action upon receipt of the written requests.

**An Evolutionary Approach to Adaptive Workflow  
Management**

**Ng Chun Fai**

**M. Phil**

**The Hong Kong Polytechnic University**

**2003**



**Pao Yue-kong Library  
PolyU • Hong Kong**

**Abstract of thesis entitled 'An Evolutionary Approach to Adaptive Workflow Management' submitted by Ng Chun Fai for the degree of M Phil at The Hong Kong Polytechnic University in November 2002.**

*Workflow technologies are associated with the control and execution of operation processes of an organization. An effectively managed workflow can mean the right information being transferred to the right person at the right time. By considering different participants of a workflow process as working together on different tasks in a business process, we propose a technique to facilitate the optimal arrangement of inter-dependent tasks.*

*Most workflow models are developed to handle static, predefined business processes. However, business operations must be adaptive to the changing business environment, and therefore, for most real-world applications, how tasks interact with one another will also have to be captured and managed. A “dynamic” workflow model that is adaptive in nature will thus be needed to cope with the constantly changing activities. In particular, since different departments within or between organizations may operate differently and may have conflicting goals, an adaptive workflow model is expected to allow a business process to be executed subject to different, possibly contradictory constraints. Given these requirements, we propose an evolutionary approach to an adaptive workflow model. The model can facilitate the handling of business activities in organizations that have to cope with rapid changes in their business environment. It uses a Genetic Algorithm to adaptively modify connections of the entry and exit points of each task so that maximum throughput, in terms of effectiveness and efficiency, can be achieved.*

*A process can be regarded as a set of connected tasks. The tasks must be connected under some precedence constraints, that is, certain tasks cannot go before some other tasks because they require information or depend on the results from those tasks. Our work optimizes the workflow process by changing the connection of the tasks while preserving the validity of the process to the precedence constraints. We denote the connections by a 2-dimensional matrix. In our evolutionary algorithm, we implement crossover by exchanging the connection configuration of randomly selected tasks from parents, hopefully combining their advantages. In addition, we implement mutation by randomly regenerating the connection configuration of randomly selected tasks to avoid the search being stuck in some local maximums. We propose a recovery mechanism to enable the*

*children from such genetic operations in our evolutionary algorithm to be valid to the precedence constraints. Minimizing the feed backward distance of tasks is the objective of our evolutionary algorithm. The objective function may be modified to allow for new optimization objectives.*

*The proposed algorithm has been integrated into a workflow-process-modeling tool. The technique is tested with simulated and real cases. The test results have shown that the proposed technique can be very useful in different real applications including those of logistics and supply chain management.*

## **Acknowledgements**

I am deeply appreciated to my project supervisor, Dr. Keith C.C. Chan, for his support, encouragement and continual guidance during the proceedings of this research work. His constructive and valuable idea and comments on this project are a great help for me. Without his supervision, I cannot taste the satisfaction in completing this project.

Special thanks go to Mr. Chester To who has provided workflow data as the test bed for the proposed framework. He also provides me the direction for solving the problem with genetic algorithm. Without him, the development of the idea would not be possible.

I would like to thank all of my colleagues who provided a nice working environment. In particular, I would like to thank Mr. Cheung King Hong, Mr. Larry Yu, Mr. Wilfred Lin and Ms Patricia Woo who have provided valuable suggestions on my research work.

Last but not least, I would like to thank my family members. This project would not have been possible without their love, understanding and encouragement.

# Table of Contents

<b>ABSTRACT .....</b>	<b>I</b>
<b>ACKNOWLEDGEMENTS .....</b>	<b>III</b>
<b>TABLE OF CONTENTS .....</b>	<b>IV</b>
<b>LIST OF FIGURES .....</b>	<b>VIII</b>
<b>LIST OF TABLES .....</b>	<b>XI</b>
<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1. WORKFLOW AND ADAPTIVE WORKFLOW .....	4
1.2. WORKFLOW FOR E-COMMERCE .....	6
1.3. DESIGN MOTIVATION .....	7
1.3.1. <i>Anticipation of fashion trend and product opportunities</i> .....	8
1.3.2. <i>Development of product feature and specification</i> .....	9
1.3.3. <i>Merchandise sourcing and procurement</i> .....	11
1.3.4. <i>Organizing production</i> .....	12
1.3.5. <i>Design and selection of merchandise distribution channels</i> .....	12
1.4. PROBLEM STATEMENT .....	17
1.5. PROPOSED SOLUTION OVERVIEW .....	19
1.6. OUTLINE OF THESIS .....	20
<b>2. RELATED WORK .....</b>	<b>21</b>
2.1. WORKFLOW MANAGEMENT .....	21
2.2. ADAPTIVE WORKFLOW MANAGEMENT .....	22
2.2.1. <i>Level of abstraction</i> .....	23
2.2.2. <i>Dynamic and Flexible Process Approach</i> .....	25
2.2.3. <i>Extended Inter-organizational Approach</i> .....	28
2.2.4. <i>Other Workflow Management Approaches</i> .....	32

2.3.	CONCLUDING REMARKS .....	33
<b>3.</b>	<b>OVERVIEW OF GENETIC ALGORITHM.....</b>	<b>35</b>
3.1.	PRELIMINARIES .....	35
3.1.1.	<i>The method</i> .....	35
3.1.2.	<i>Encoding</i> .....	37
3.1.3.	<i>Genetic Operations</i> .....	38
3.1.4.	<i>Steady State GA</i> .....	39
3.2.	GENETIC ALGORITHM AND GRAPH PROBLEM .....	39
3.3.	PROPOSED GRAPH OPTIMIZATION BY GENETIC ALGORITHM .....	41
3.4.	DIFFERENCE BETWEEN JOB-SHOP SCHEDULING PROBLEM AND PROPOSED WORKFLOW PROCESS OPTIMIZATION PROBLEM.....	44
3.5.	REMARKS.....	45
<b>4.</b>	<b>ADAPTIVE INTER-ORGANIZATIONAL WORKFLOW – EVOLUTIONARY APPROACH</b> <b>46</b>	
4.1.	OVERVIEW .....	46
4.2.	SEGMENTATION OF INTER-ORGANIZATIONAL PROCESS ACCORDING TO ORGANIZATIONAL BOUNDARY .....	47
4.2.1.	<i>Definition of segmentation</i> .....	48
4.2.2.	<i>Requirement of segmentation</i> .....	48
4.3.	EVOLUTIONARY ALGORITHM FOR WORKFLOW OPTIMIZATION .....	50
4.3.1.	<i>Overall algorithm</i> .....	50
4.3.2.	<i>Model of a process</i> .....	50
4.3.3.	<i>Chromosome representation of workflow process</i> .....	56
4.3.4.	<i>Initial Population Generation</i> .....	56
4.3.5.	<i>Genetic Operators Implementation</i> .....	63
4.3.6.	<i>Fitness and Stopping Criterion</i> .....	69
<b>5.</b>	<b>EXPERIMENTAL RESULTS.....</b>	<b>72</b>
5.1.	THE EXPERIMENT DETAILS .....	72

5.1.1.	<i>Experimental Data</i> .....	72
5.1.2.	<i>Objective of Experiment</i> .....	73
5.1.3.	<i>Experiment Design</i> .....	73
5.2.	INTRODUCTION TO GREEDY APPROACH .....	73
5.3.	PERFORMANCE COMPARISON .....	74
5.3.1.	<i>Simulated data</i> .....	74
5.3.2.	<i>Real Data</i> .....	78
5.4.	DISCUSSION .....	81
5.4.1.	<i>Advantage in performance over greedy approach</i> .....	81
5.4.2.	<i>Influence on performance by process size</i> .....	81
5.4.3.	<i>Influence on performance by constraint</i> .....	81
<b>6.</b>	<b>SYSTEM DESIGN AND IMPLEMENTATION</b> .....	<b>82</b>
6.1.	OVERVIEW OF THE WORKFLOW PROCESS MODELING TOOL – THE IMPLEMENTATION PLATFORM	
	82	
6.1.1.	<i>Process</i> .....	84
6.1.2.	<i>Task</i> .....	86
6.1.3.	<i>Limitation on original implementation</i> .....	87
6.1.4.	<i>Concluding remarks</i> .....	87
6.2.	ADAPTIVE WORKFLOW FEATURE ON PROCESS MODELING TOOL .....	88
6.2.1.	<i>The workflow data</i> .....	88
6.2.2.	<i>Initial Optimal Process Generation</i> .....	90
6.2.3.	<i>Adaptation to changes in constraints for existing process structures</i> .....	94
<b>7.</b>	<b>CONCLUSION</b> .....	<b>103</b>
7.1.	SUMMARY .....	103
7.1.1.	<i>Unique feature of the proposed framework</i> .....	104
7.2.	FUTURE WORK .....	104
7.2.1.	<i>Application on e-commerce application</i> .....	104
7.2.2.	<i>Use of agents</i> .....	106
7.2.3.	<i>Data Warehouse and Knowledge Management</i> .....	106

7.2.4.	<i>Application on Supply chain management</i> .....	106
<b>8.</b>	<b>REFERENCES</b> .....	<b>108</b>

# List of Figures

Figure 1.1 Dimension of workflow management issues.....	5
Figure 1.2 Process of fashion product portfolio planning.....	10
Figure 1.3 Activities of fashion product innovation for global marketplaces.....	13
Figures 1.4 Information dependency and feed-backwards .....	14
Figure 1.5 Feed-backward loops.....	15
Figure 1.6 Shorter Feed-backward loops .....	16
Figure 2.1 Abstraction levels of workflow models .....	24
Figure 2.2 Inter-organizational workflow process .....	30
Figure 2.3 Integration of proposed framework .....	34
Figure 3.1 The genetic algorithm.....	37
Figure 3.2 Parent Selections.....	38
Figure 3.3 Steady-State Reproductions.....	39
Figure 3.4 Example of process as connection of nodes .....	42
Figure 3.5 Corresponding connection matrix representation for the process .....	42
Figure 3.6 Example of undirected graph.....	43
Figure 3.7 Connection matrix for an undirected graph.....	43
Figure 4.1 Tasks before optimization.....	46
Figure 4.2. Segmentation .....	47
Figure 4.3 Recombine after optimization .....	47
Figure 4.3. Pseudo code for genetic algorithm .....	50
Figure 4.4 A Process example.....	51
Figure 4.5 Constraint rules.....	53
Figure 4.6 Information carriages of tasks .....	53
Figure 4.7 Information carriage of another process.....	54
Figure 4.8 Feed-backward.....	55
Figure 4.10 Tasks before optimization.....	58
Figure 4.11 Information dependency for Task D .....	58
Figure 4.12 Candidates for proceeding task of Task D.....	59
Figure 4.13 Assignment of connection from Task B to Task D .....	59
Figure 4.14 Information dependency for Task D .....	59
Figure 4.15 Assignment of connection from Task A to Task D .....	59

Figure 4.16 Redundant Link Example .....	60
Figure 4.17 Information dependency for Task B .....	60
Figure 4.18 Candidate for preceding task of Task B.....	60
Figure 4.19 Assignment of connection from Task A to Task B.....	61
Figure 4.20 Redundant Link Removed .....	61
Figure 4.21 Candidates for preceding task for Task C.....	61
Figure 4.22 Linked assigned to Task C .....	61
Figure 4.23. Finished result.....	62
Figure 4.24. Selection .....	64
Figure 4.25. Crossover .....	64
Figure 4.26. Selected parents .....	65
Figure 4.27 Crossover – Swap of connection configurations for selected tasks.....	66
Figure 4.28. Recovery algorithm .....	66
Figure 4.29. Recovery of children in reproduction .....	67
Figure 4.30. Mutation – regenerate configuration for selected tasks .....	67
Figure 4.31. Mutation algorithm .....	68
Figure 4.32 Feed back path length measurement.....	69
Figure 4.33 Longest Path .....	70
Figure 5.1 The pseudo code for greedy algorithm .....	74
Figure 5.2 Searching performance for simulated data of 103 tasks and complexity of 46.....	75
Figure 5.3 Searching performance for simulated data of 103 tasks and complexity of 86.....	76
Figure 5.4 Searching performance for simulated data of 103 tasks and complexity of 125.....	77
Figure 5.5 Searching performance for real data of 36 tasks and complexity of 197 .....	78
Figure 5.6 Searching performance for real data of 35 tasks and complexity of 245 .....	79
Figure 5.7 Searching performance for real data of 31 tasks and complexity of 14380 .....	80
Figure 6.1 Workflow Modeler Interface .....	83
Figure 6.2 Panel .....	84
Figure 6.3 Task properties dialog box.....	84
Figure 6.4 Process structure .....	85
Figure 6.5 Task properties for A244 .....	86
Figure 6.6 Information dependency constraints.....	90

Figure 6.7 Tasks before optimization.....	91
Figure 6.8 Generating the optimal structure .....	92
Figure 6.9 Generated optimal process structure.....	93
Figure 6.10 Tasks affected in change .....	97
Figure 6.11 Modification of properties for tasks .....	98
Figure 6.12 Modification for properties of task A244 .....	99
Figure 6.13 Modification for properties of task A244 .....	99
Figure 6.14 Modification for properties of task A245 .....	100
Figure 6.15 Modification for properties of task A245 .....	100
Figure 6.16 Re-generation of optimal process structure after changes in constraints .....	102

# List of Tables

Table 2.1 Issues covered by literature in adaptive workflow management .....	33
Table 3.1 Similarity among job shop scheduling and process optimization .....	44
Table 3.2 Differences among job shop scheduling and process optimization .....	44
Table 4.1 Experiment Tasks for the algorithm .....	63
Table 4.3 Resource taken for each alternative path.....	70
Table 4.3 Resource taken for each alternative path.....	71
Table 5.1 Searching performance for simulated data of 103 tasks and complexity of 46.....	74
Table 5.2 Searching performance for simulated data of 103 tasks and complexity of 86.....	75
Table 5.3 Searching performance for simulated data of 103 tasks and complexity of 125.....	76
Table 5.4 Searching performance for real data of 36 tasks and complexity of 197	78
Table 5.5 Searching performance for real data of 35 tasks and complexity of 245	79
Table 5.6 Searching performance for real data of 31 tasks and complexity of 143	79
Table 6.1 Functions description .....	84
Table 6.2 Sample tasks data .....	89
Table 6.3 Change in information dependence as there are changes in business environment .....	96

# 1. Introduction

The E-Commerce infrastructure is concerned with the ability of performing business online via the Internet. Businesses are not constrained to geographic locations any more. The Internet infrastructure provides a means for people to communicate with the world. Customers may contact companies all over the world for business transactions. For this reason, companies have to collaborate with customers and suppliers worldwide. The existence of the Internet and WWW provides the hardware and communication infrastructure for people to do business over the multi-organization environment in a relatively inexpensive manner. It is important to control and coordinate business collaboration with a comprehensive software infrastructure.

In an effort to coordinate and consolidate a large number of global activities and communication processes while maintaining regional responsiveness and autonomy, a lot of international fashion distributors and buying companies attempt to manage their global activities from the perspective of an integrated organization by sorts of novel methods or tools. One such effort is to plan and model the process system holistically so that the information and opinions at all levels of an enterprise can be shared, discussed and revised effectively and collaboratively at the early phase of product conception and embodiment, and the minimum changes in the later stages of production and distribution can correspondingly be entailed. On such premises, the planning and modeling of the processes should be based on an approach that emphasizes integrative and effective process sequencing among the interacting activities. Extant theoretic concepts in managing new product development posit that employing activity-process-flow modeling-paradigm. Such paradigm and perspective gives rise to achievement of several

goals:

- To identify how and when information is provided to a point of activity processes so that the process can be executed sufficiently and completed in time.
- To ensure a high level of accuracy and concurrency for activity interaction and responsive identification of the essential attributes of 'winning' products at the last possible moment in the course of new product development. Such winning products are highly desirable in the market and best meet changing customer requirements;
- To schedule short life cycle products to markets in a timely manner so that the profit potential of new products and market uncertainty can be reasonably anticipated;
- To aid in sequencing activity processes so that undesirable iteration for information revision and review can be avoided and controlled; or the costly product change at the later production stage can be anticipated and managed;
- To provide a compact and highly transparent representation of the interactive process or communication. With such representation, teams and facilities can be more easily managed and consolidated in terms of resource and response time ;
- To generate a descriptive model of the activity system that provides a prescription for continuing organizational learning and improvement in project management.

In our research work, we attempt to put forth these concepts to research an international fashion and textiles buying company that is aiming at utilizing new methods for

improving process planning and coordination of its global activities. At the initial research stage, the company representatives discuss with the authors about the issues they face in planning and organizing the activities for new product development and implementation in the global market environment. A key issue they raise is the problem of intractable interactions among a large number of interdependent activities that make communication difficult to manage. Very often, information after passing through a number of subsequent activities is detected and recognized to be inadequate, incorrect or highly uncertain. It forces product development teams to call for upstream activity processes to re-examine and adjust the information. This is attributable to the facts that no single team or enterprise is able to perceive all the perspectives throughout all phases of a product life cycle and anticipate all the possible influences by external environmental market factors.

To illustrate the significance of this in the global fashion businesses, the company cites one common example in which the marketing teams capriciously ask design teams to revise the design of new products in the upstream processes in anticipation of probable changes in market demand or affordability. After the design teams revise the design idea and conceptions, the manufacturing teams correspondingly adjust the design of production setup and re-schedule raw material distribution and financial arrangement. Such changes would in turn necessitate the marketing teams to re-consolidate the end-market operations in view of global image, pricing and quality consistence. Inefficiency and ineffectiveness of activity communication and interactions at the stage of new product development would result in a big change at the later stage of product manufacturing and distribution, incurring significant cost despite having the problems un-resolved. Managing interactions among interdependent activities becomes the key to

effective and timely new product design and development.

Workflow technology provides a software infrastructure to support people to work together to achieve a common goal inside an organization, which is important for E-Commerce applications. Workflow has to be extended so that it is capable of supporting collaborative activities over a multi-organization environment for E-Commerce Applications.

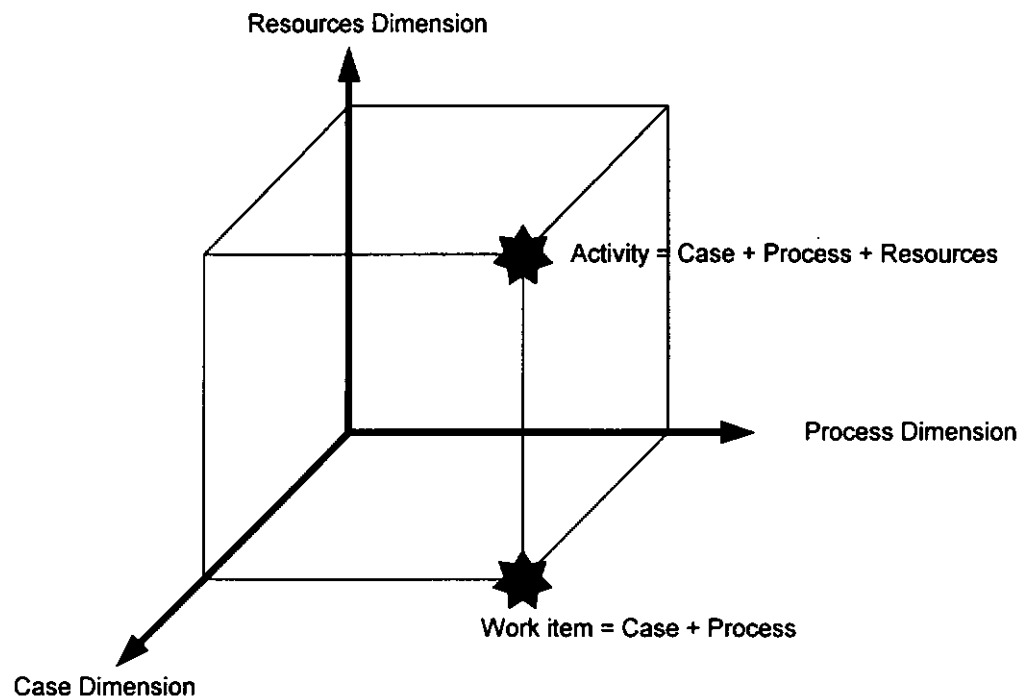
Before at our proposed solution, we would like to have an overview on the different aspects of the workflow technology.

## **1.1. Workflow and adaptive workflow**

Workflow management refers to the domain, which focuses on logistics of business activities. It can be defined as a computerized facilitation or automation of a business process, in whole or part. [WfMC 96] The ultimate goal of workflow management is to deliver the right information to the right person at the right time and ensure these people execute proper actions at that time.

Business activities can be identified by cases. Each workflow instance is specifically for a case. Workflow management aims to handle cases effectively and efficiently. Tasks refer to sub-part of work done for a case. Executing tasks in a specific order fulfills this objective. How the tasks are ordered is known as process definition. There may be many cases that can be handled by a specific ordering of tasks. For this reason they share the same process definition. The entity that executes the tasks is known as a resource.

We can see that workflow has three dimensions, namely case dimension, process dimension and resource dimension. These dimensions can be visualized in Figure 1.1.



*Figure 1.1 Dimension of workflow management issues*

Workflow technology has the following aspects:

- Routing of information/documents
- Allocation of tasks to actors
- Scheduling of tasks in time
- Scheduling of scarce resources
- Monitoring flow of work
- Handling exceptional situations

- **Providing Management information**

As cases may differ from one another and resources availability changes from time to time, it is obvious that a static process definition is not fully compatible with each case. Static mapping from resources to tasks also cannot fulfill the actual environment all the time. Adaptive workflow aims to provide flexibility to workflow management by supporting changes in process definition. The need of flexibility for current workflow management technology is widely recognized.

## **1.2. Workflow for E-Commerce**

Broadly speaking, e-commerce means performing selling and buying of products or services electronically. It is widely perceived that being done electronically means being done remotely over the Internet by Electronic Data Interchange (EDI), which plays an important role in many business activities in E-Commerce such as transaction processing and confirmation. The implication is that such activities no longer constrained to geographical and organizational boundary and inter-organizational business process are more difficult to be separated from intra-organizational ones. [Van der Aalst 99] proposed the process aspect of E-Commerce by relating it to workflow management. Traditional workflow technology must be extended to support the modeling of inter-organizational business activities before being applied to E-commerce applications. An inter-organizational workflow model provides the foundation to e-commerce application, which must be capable of operating in a distributed environment involving multiple parties with dynamic availability and a large number of heterogeneous information sources with evolving contents.

Current research work for inter-organizational workflow concentrates on extending the

workflow model to incorporate capability for cross-organizational operation support. The execution efficiency on process operation under such full-of-change environment is seldom considered. The inter-organization process often incorporates iterations, each step of which depends on information from other steps. Such iterations may repeat previous work done since the information on which this work depends might have been changed in other steps. Such repeated work incurs additional costs to the overall process, in terms of time and other resources and hence affects business performance. Such performance degradation can be avoided or minimized by re-arranging the activities.

### **1.3. Design Motivation**

Workflow technology organizes business activities into structural processes and provides management and control to the execution of these processes. Such processes consist of tasks that are inter-connected. When a task is completed, the execution proceeds to the next task connected to it. In the real world, the order of task connection is usually governed by certain constraints. Hence the tasks cannot be connected in an arbitrary order. Because how some tasks are performed usually depends on information from other tasks, such tasks cannot be executed before the tasks on which they depend are finished. Such information dependency constrains the order of tasks execution, and hence the process structure.

To illustrate the situation clearly we use a scenario in the fashion design industry. The activities are based on the research results studied in the international fashion and textiles buying company, which supports its existing worldwide fashion stores in product development and sourcing. In the global fashion marketplace, product innovation involves flows of information and opinions across a number of functional teams and cross-country enterprises that collaboratively interpret market opportunities and

requirements and transforming them into a set of technology assumptions about product features and utilities. In general, the innovation comes across several distinct processes, (1) anticipating fashion trend and product opportunities; (2) developing product program and specification; (3) sourcing and allocating fashion merchandise procurement; (4) organizing production; and (5) arranging merchandise shipment.

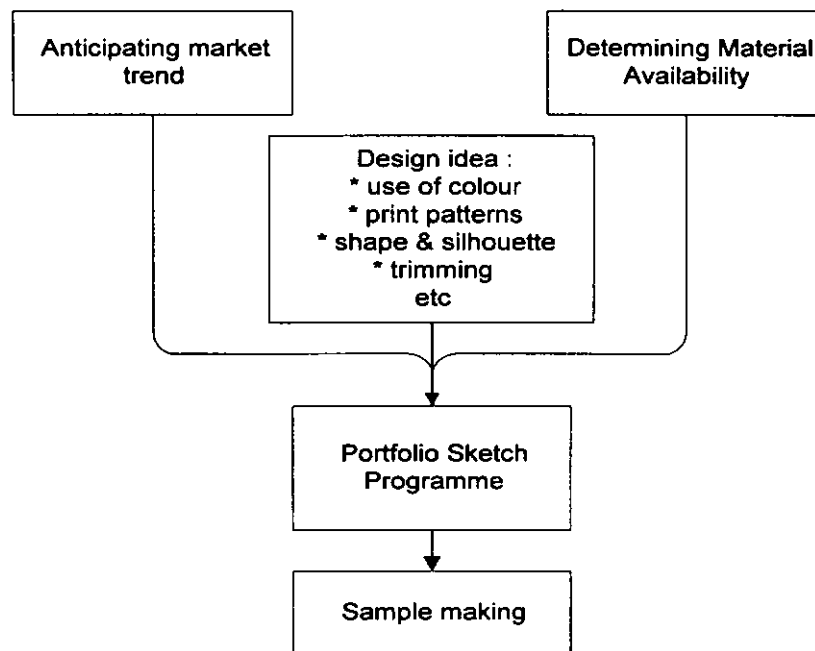
### **1.3.1.Anticipation of fashion trend and product opportunities**

Anticipation of such trend and opportunities involves market information search and analytical planning to understand the concurrent market preferences and the major factors of fashion adoption. This understanding is to foresee what fashion concepts will be accepted in coming seasons. The anticipation includes observation of general market-wide developments and the environmental opportunities for fashion innovation. Furthermore, the process focuses on the characteristics of existing fashion products, and their strengths and weaknesses on the basis of market popularity. Indeed the activities in this phase define not only the product specifications and the basic physical features, but also the augmented product values that will be perceived in customers' mindsets. A useful expression interpreting such product values includes a set of attributes, like fashion aesthetic appealing, social conformity, technical performance appropriateness, perceived quality, convenience and so on. Such information is very abstract and conceptual. Because of this, the coordination across cross-country functional teams and enterprises is much characterized by a high level of informal and context-specific communication. Process effectiveness is therefore to a great extent determined by how the activity processes are interfaced, and how well and how easy the information is transmitted and

understood. During this process of product innovation, communication is very ambiguous and vulnerable. [Eckert et al. 00], [Jassawalla & Sashittal 98] and [Milne 00] offer comprehensive discussions about the issues and associate them with techniques proposed to capture and trace the interaction amongst innovation teams.

### **1.3.2. Development of product feature and specification**

Following the identification of market trends and preferences at the previous stage of the process chain, development teams comprising designers, marketing teams, and technical teams, attempt to define the criteria to develop the portfolio and product samples in terms of fashion features that appeal to customers' choice. This comprises the choice of colours, cutting, silhouette, texture, trimming adoption and workmanship. A new seasonal design collection is sometimes compiled as a portfolio that is established to extend the existing product line, as shown in Figure 1.2. In practice, a variety of portfolios are developed and screened for a season to select a final one consistent with the company's objectives.



*Figure 1.2 Process of fashion product portfolio planning*

In brief the processes at this stage are very methodological and have to be well controlled. Product innovation in global marketplaces stems from recognition of market needs and interpretation of such needs into product carrying respective attributes that can convey product values to customers. Logically thinking, product innovation is guided by aggregate customer consumption patterns, which are evolutionary in nature and changing in pace with all other socio-technological aspects. [Regan et al 97] survey and analyze in much detail the difference and similarity between the engineering design process and the apparel design process, both empirically and literally. The results reveal the facets to the innovation for fashion and textile products are tractable more often than not and methodological, and its success is much determined by how detailed the innovation can be specified and communicated among all teams involved. [Plumlee & Little 98, Mills,

### **1.3.3.Merchandise sourcing and procurement**

In this process the innovation activities include the search and selection for sourcing possibilities, which are in principle able to produce novel product responsively and agilely. The critical consideration during the process involves how to optimize profit, assure the least uncertainty in the supply market, schedule timely delivery and arrange the financing aspects. As observed, sourcing and procurement refer to activities of purchasing, storing, handling traffic, receiving and inspecting incoming material and salvage. The teams should anticipate and be alert to various sourcing restrictions and the methods to allocate the bases of supply according to predetermined sets of business criteria and company objectives. Very often, activities in this stage involve material management and order placement to achieve a timely production schedule. In brief, they include:

- researching information of supply market, trade or legal restriction (like quota control or tariff variations);
- checking requisition;
- analyzing quotations of both material supply and the manufacturing process;
- evaluating and choosing suitable suppliers;
- scheduling delivery and order placement;
- negotiating and writing orders;
- checking regulatory conditions of trade;

- following up for delivery;
- verifying invoices;
- corresponding with suppliers and buyers.

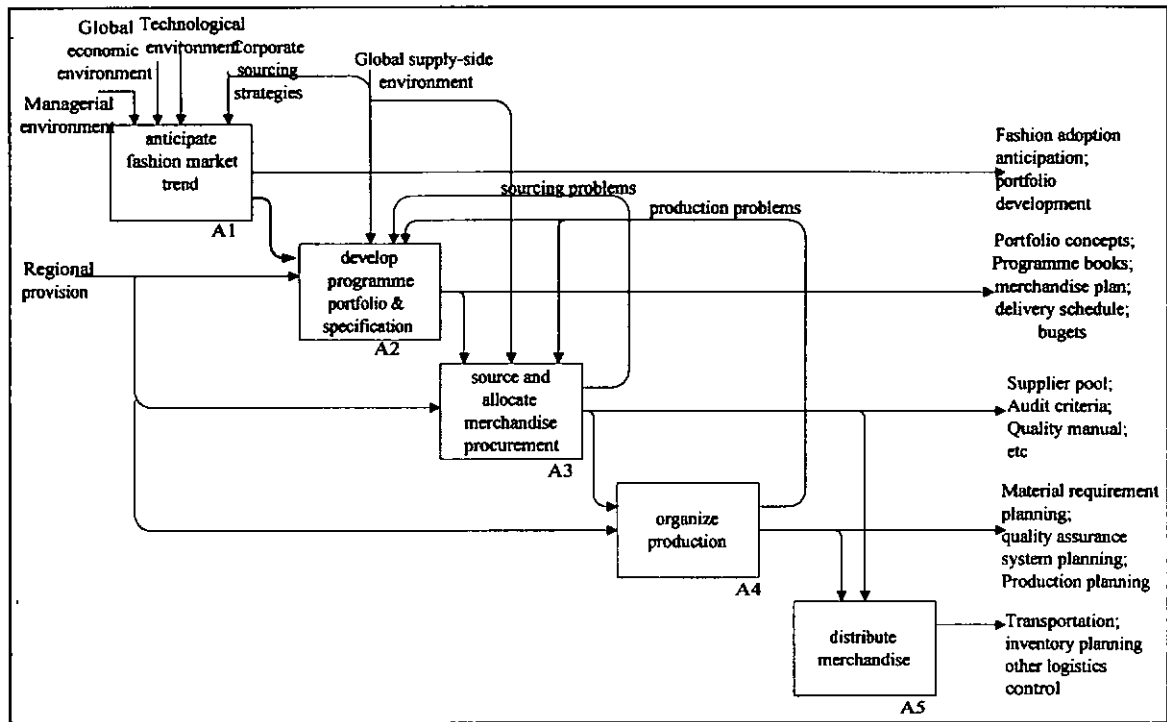
#### **1.3.4.Organizing production.**

This stage of the process-chain emphasizes how to arrange production, define quality level, make schedule of material requirement and delivery, and allocate control personnel. The process involves activities of expediting material orders and receipts, inspection, resolving technical problems and arrangement of finished product packaging and dispatch.

#### **1.3.5.Design and selection of merchandise distribution channels**

These last stage activities are to examine distribution channel member performance and decide holistically the logistic systems. It involves planning and orchestrating transportation methods and routes, packaging and handling methods, and finally advising the best arrangement of delivery to all the other parties concerned.

Using our framework, we establish an IDEF0 flowchart to represent the above observed innovation process and model the product innovation activity structure and relationships meticulously, as shown in Figure 1.3.

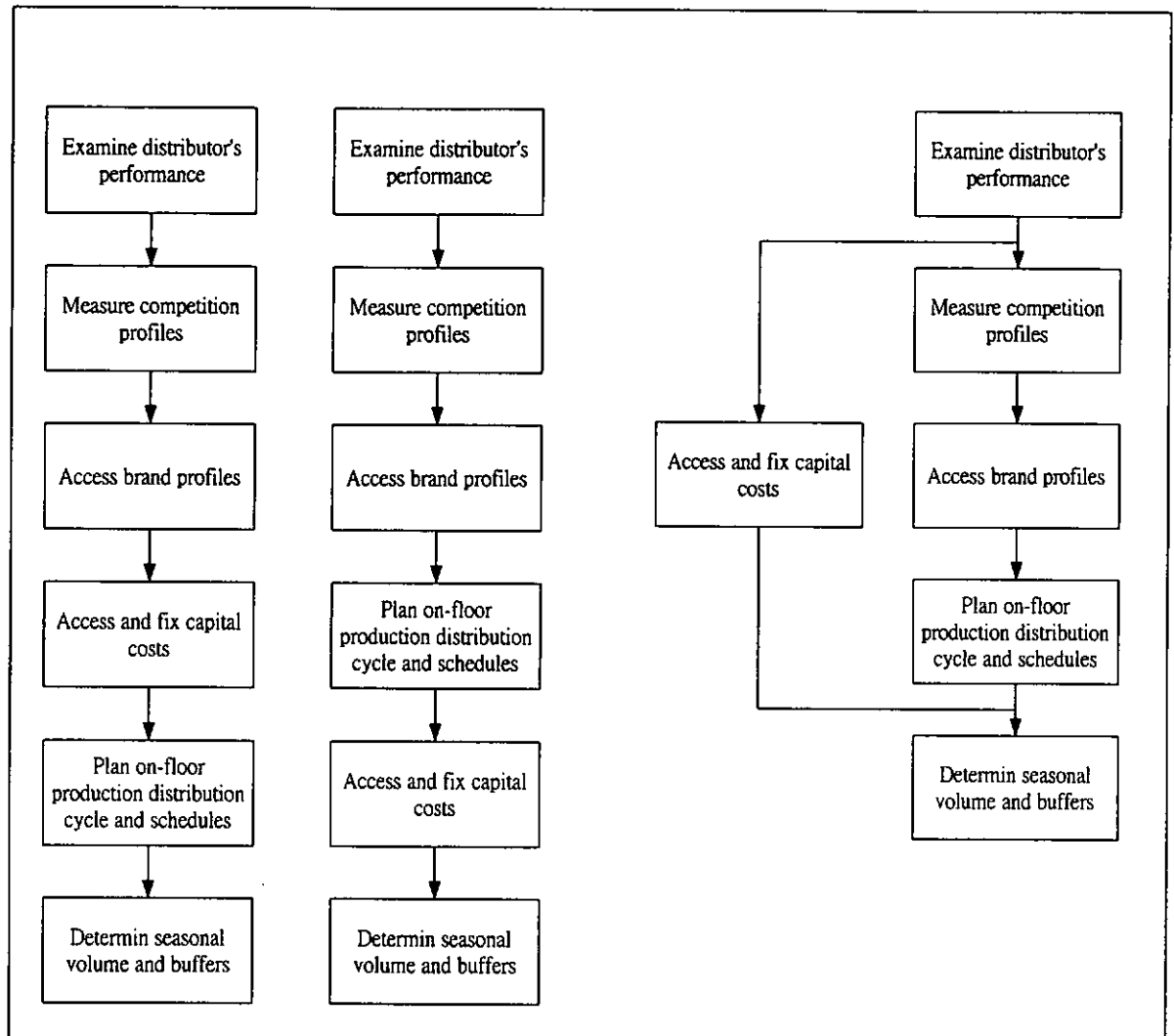


*Figure 1.3 Activities of fashion product innovation for global marketplaces*

The flowchart corresponds with the structure of activity interdependency that is set forth in the course of new product innovation and development. This workflow process can be decomposed structurally into more detailed levels of an activity hierarchy, based on a top-down analytical breakdown approach. Such decomposition provides the gradual exposition of details by expanding the contents of the activity boxes on one level into deeper and deeper levels. The box numbers act as convenient indices to search for levels of exposition of activity tasks. Thoroughly indexed activities researched in the case are listed in Table 5-5. As stated, there are 103 decomposed activities operated by a number of technical and functional teams in different locations. A detailed listing of these tasks and their information dependency relationship could be found in the Appendix.

In the following example tasks “Plan on-floor product distribution cycle and schedule” and “Access brand profile” are dependent on information from task “Measure

competitors profile”. For this reason, these two tasks cannot be executed before task “Measure competitors profile”. Under such constraints, possible process structures are as follows:

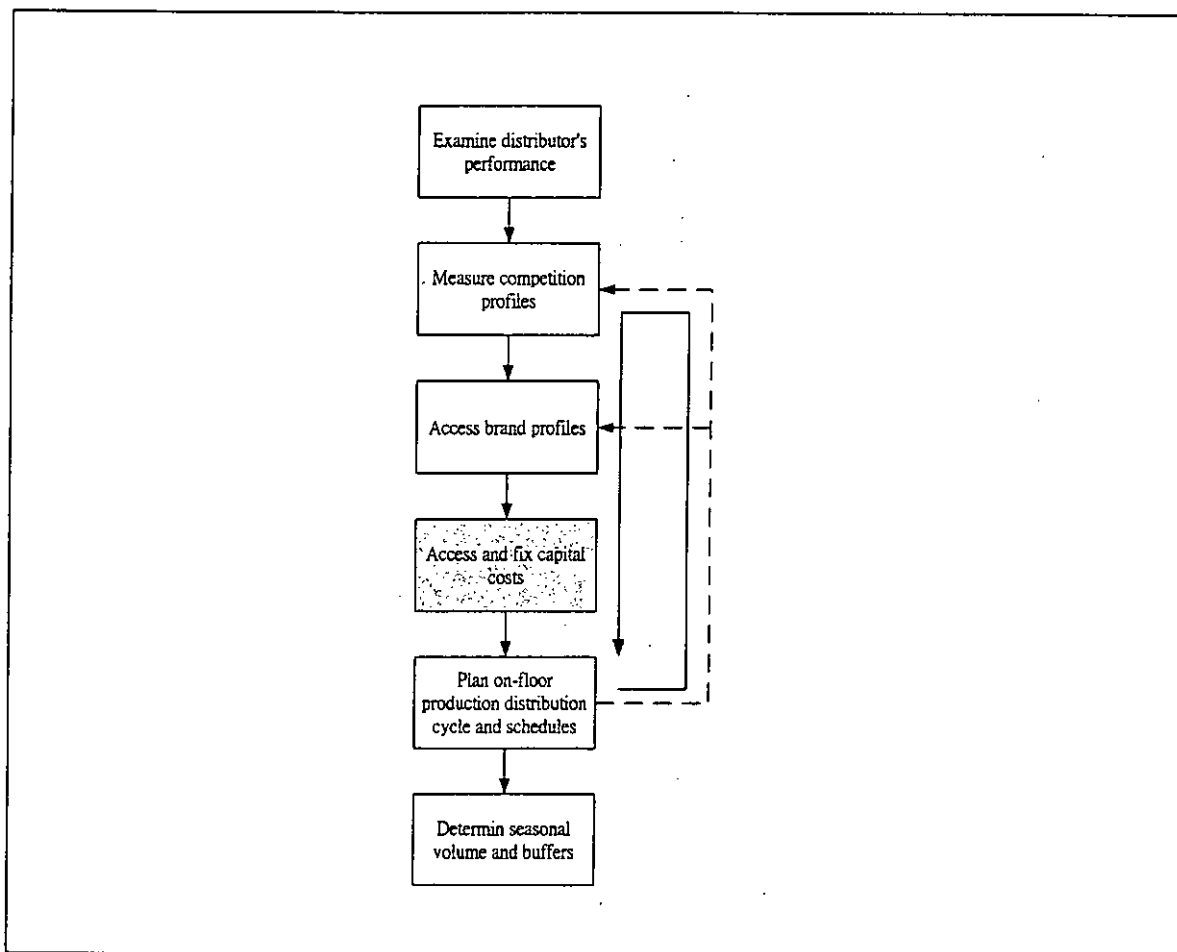


*Figures 1.4 Information dependency and feed-backwards*

The dependency described concerns the normal flow of an execution. There is another type of dependency. Due to exceptional cases such as unsatisfactory result or requests for change, the flow of execution may have to roll back to some previous tasks. And the tasks between the original and destination of rolled back have to be repeated. We call the relationship between origin and destination of roll back information feed-backward. In fact, this is another type of information dependency as the destination of feed-backward

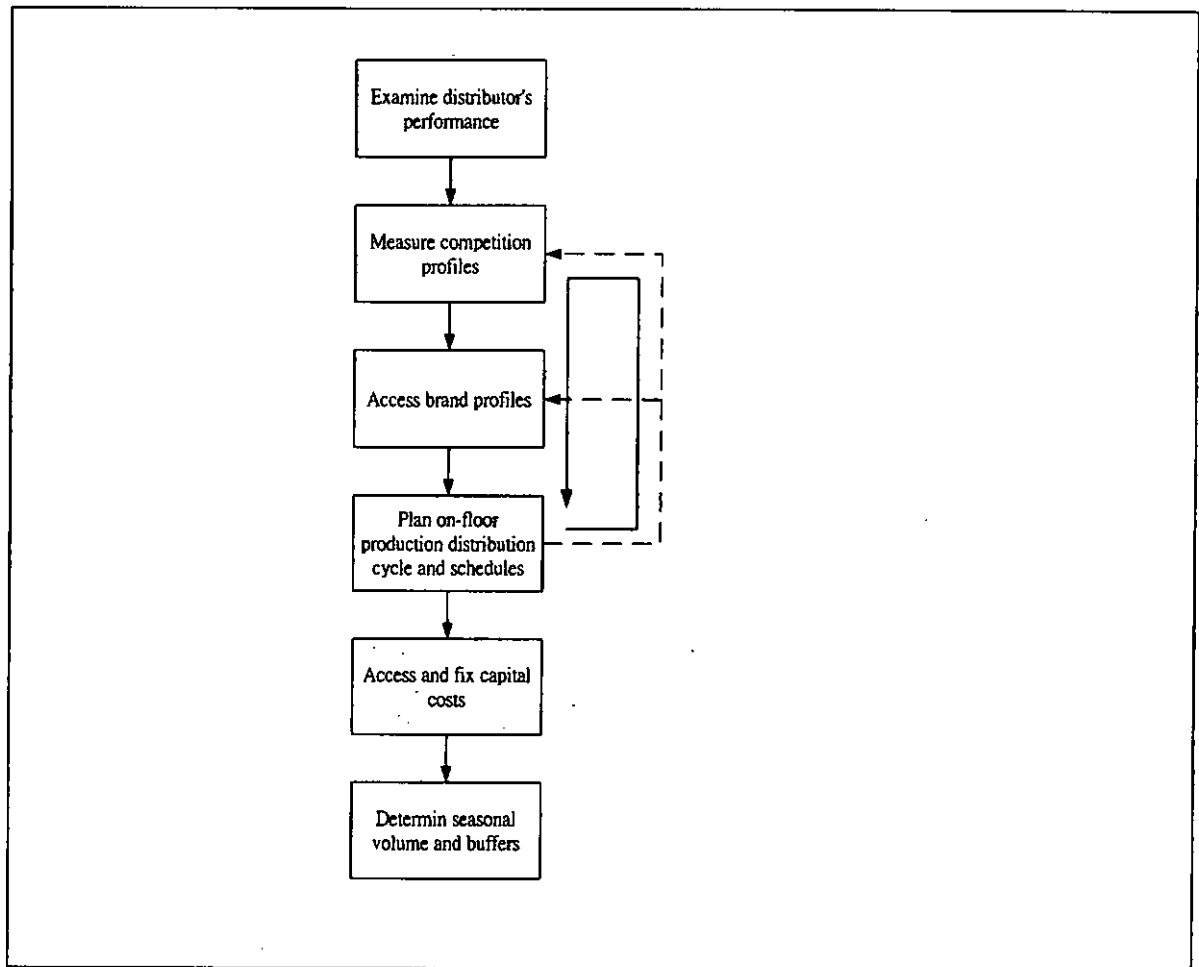
depends on information from the origin. The repeated tasks section of process forms an iterative loop, which may be repeated one or more times until a final satisfactory result comes out. It is usual for business activities.

Improper arrangement of tasks may make some irrelevant or unnecessary tasks get inside such loop. This incurs additional costs to overall business performance since the irrelevant parts of the process are repeated with the iteration loop.



*Figure 1.5 Feed-backward loops*

If we re-arrange the tasks order, the repeat of irrelevant works could be eliminated.



*Figure 1.6 Shorter Feed-backward loops*

From the above example, we believe that proper re-arrangement of tasks inside a process for business activities should improve efficiency and performance.

## 1.4. Problem Statement

Though workflow technology had been extended to support inter-organizational activities, there are rooms for improvement. While concentrated on model capability, current solutions do not focus on efficient execution.

Workflow enactment in the intra-organizational environment ensures each party achieves the same goal. The inter-organizational activities can be managed as controlled processes. These processes often contain iterations, which end only if a satisfactory final result comes out. Before such result comes out, the iteration based on the improper arrangement of operation incurs additional time and resources for repeatedly fixing unacceptable/unsatisfactory results. If activities among organizations can be properly arranged, the amount of repeated work can be minimized and the additional cost spent on repeated work minimized.

Under all workflow models, business operations are processes consisting of tasks. In the process structure, each task carries information from its preceding tasks. Preceding tasks of a task are those that have either a direct or indirect connection to it. We use  $Pr$  to denote such relationship. A process  $P$  can be defined as the following:

$$P = (T, E, R)$$

$$T = \{T_1 \dots T_i, T_j \dots T_n \mid i, j \in \{1 \dots n\}\}$$

$$E = \{Pr_{ij} \mid i, j \in \{1 \dots n\}\} \text{ where } Pr_{ij} = Pr(T_i, T_j)$$

$$\text{If } Pr_e(T_i, T_j) \text{ and } Pr_e(T_j, T_k) \text{ then } Pr_e(T_i, T_k)$$

Where  $T$  is a set of  $n$  Tasks of the process

$E$  = Set of connection among tasks

$Pre$  = Precedence Relationship among tasks

Precedence constraint defines the necessary information required for each task. All tasks

are dependent in nature. For this reason, if a task is dependent by other tasks, it must precede all of those tasks. If a precede relationship holds for these dependent tasks, the constraint is stratified. Such facts can be denote symbolically by following:

$$R = \{R_{ij} \mid i, j \in \{1 \dots n\}\} \text{ where } R_{ij} = R(T_i, T_j) \\ Pre(T_a, T_b) \Rightarrow R(T_a, T_b)$$

For a process to be valid in overall, all constraints must be fulfilled. That means the precedence relationships among all tasks must cover all constraints requirement.

$$Pre \supseteq R$$

Inside the process structure, each tasks may have feed-backwards to its preceding task.

We denote the feed-backwards as the following:

$$F = \{F_1 \dots F_l\} \\ \forall x \in \{1 \dots m\}, F_x \Rightarrow Pre(T_c, T_d) \text{ and } F(T_d, T_c), \text{ where } c, d \in \{1 \dots n\}$$

Such feed-backwards will cause delay and incur extra costs. As all preceding tasks had to be repeated, we define the duration of the delay from the point of feed-back again. If there are multiple paths between the feed-backward pair, we will assume the cost is the path with the maximum cost.

$$V(Pre(T_a, T_b)) = V(Pre(T_a, T_0)) + 1 \text{ for some } T_0 \\ \text{such that } [T_a = T_0 \text{ or } Pre(T_a, T_0)] \text{ and } V(Pre(T_a, T_0)) \text{ is maximum}$$

We will try to find a connection set such that the total cost for the feed-backwards of the process is minimum, provided that the process structure still is valid. The operation can be illustrated as following:

$$\sum V(Pre(T_a, T_b)) \text{ such that } F(T_a, T_b)$$

In summary, the problem we are concerned with can be stated as follows. Given a set of tasks and a set of constraints, we try to find out the process configuration, which precedence relationship confines to the constraints requirement and the repeating task incurred by feed-backwards is minimized.

The proposed solution to this problem will automate the process of ordering, that is capable of adapting to changing constraints. Allowing accurate and objective evaluation of its efficiency. It will be a fundamental impact to the modern business environment.

## 1.5. Proposed solution Overview

The problem of identifying a task arrangement, which is the most efficiency in execution, is an optimizing searching problem. We use genetic algorithm to optimize the inter-organization workflow by finding out the optimal arrangement of tasks inside the process structure. The main advantage of using genetic algorithm is that it can potentially accommodate new consideration for optimization by altering the design of the objective function. Experimental result has shown that the approach provides positive results.

We use the problem downsize approach to solve the problem. For the inter-enterprise environment, it makes sense to optimize the operation of each entity, then combine the results of each entity to give the whole picture.

In our evolutionary algorithm, each workflow process is encoded in chromosome in the form of a two-dimensional array, which indicates the connection graph of tasks in that process. Initially we generate a set of feasible process structure randomly as our initial population. We randomly select pairs of chromosome and swap the genetic information, which is the connection configuration of randomly selected tasks for each process. After

such swap, we add connections to recover the broken information dependency, if any, during the swap. We may also randomly change the connection configuration of some tasks in some process structures. Under such operations, we hope that some new process structures better than that previously generated can be produced. We stop the algorithm and arrive at a solution, which is the best chromosome in the population, when the fitness of the population converges to a certain extent.

## **1.6. Outline of Thesis**

This thesis is structured as follows. In section 2 describes current related work done on using workflow technology to solve e-commerce application problems. In section 3 we will have an overview of the evolutionary algorithm, which is the basis of our solution. In section 4 we propose our solution to optimizing the workflow process by the evolutionary algorithm, and gives the implementation details of the solution. In section 5 we gives experimental results. In section 6 we introduce a system design and implementation of the proposed solution. Section 7 concludes the thesis.

## **2. Related Work**

Before we go ahead to our literature review, we will first have a look at the architecture issues of a comprehensive solution for workflow management. Then we will have a look at how the work done in literature fulfills the requirements of these issues and their inadequacy. Last, we will identify how our work complements this inadequacy.

### **2.1. Workflow Management**

[Schulz and Orłowska 01] stated the issues on architecture of a Workflow Management System (WfMS):

- Generic product structure
- Process definition
- Process definition meta model
- Client applications
- Invoked applications
- Interoperability
- Administration and monitoring

Generic product structure defines what the components of a WfMS should be and how they interact or communicate. Process definition gives the formats and interfaces in which business processes are defined. Process definition meta-model defines the top-level entities contained in a process definition. It also defines the convention for

grouping processes into a related model and the use of common definition data across a number of different process definitions or models. Client applications concern the application-programming interface (API) for WfMS user clients. Invoked application concerns the API for interaction between WfMS and associated applications. Inter-operability provides an abstract specification that defines the functionality necessary to achieve a defined level of interoperability among two or more workflow engines. Administration and monitoring provide the facilities for control and keeping track of the progress and status of workflow instance, and perform corrective actions upon exceptional cases.

These issues point out the things that should be considered when designing a comprehensive workflow management solution. It is merely a guide and no concrete implementation is provided. Some pieces of work in literature try to respond to these issues, but we have not yet seen a complete one.

## **2.2. Adaptive workflow management**

A workflow reference model only defines a framework for various WfMS implementations. Some traditional implementations with a static process definition are often criticized for their “rigid” workflow models, which sometimes require manual coordination outside the system to carry out the task forced upon the user by the system. This issue introduces the need for an adaptive workflow approach. Adaptive workflow implementation provides flexibility to workflow modeling which makes it possible to adapt the actual execution environment of the process. Such feature is essential for a large-scale workflow management system of an enterprise in a distributed environment.

The architecture issues stated in section 2.1 are fundamental to a workflow management

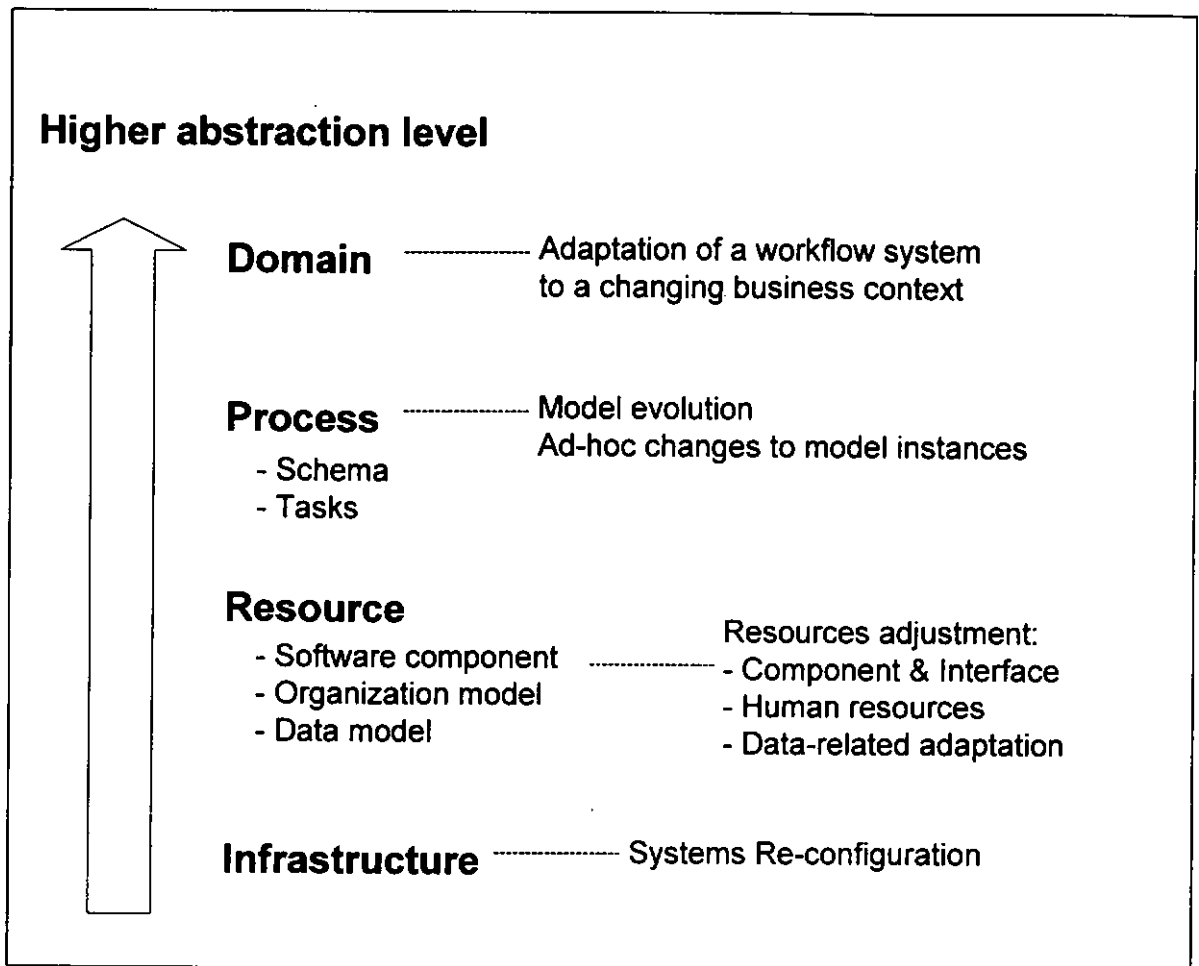
system structure. Based on this, [Han, Sheth and Bussler 98] addressed the need of adaptive workflow.

Evolution due to changing environment - Business activities and environment are highly dynamic and subject to continuous change. For this reason, re-structuring and optimization of existing business process will become indispensable for most effective and efficiency operations.

Ad-hoc derivation during workflow execution, usually dynamic refinement, user involvement, unpredictable events and erroneous situations, will cause the workflow instance to perform ad-hoc changes. Such changes usually do not affect the workflow schema. [Tang 99]

### **2.2.1.Level of abstraction**

Proper ways of handling workflow adaptation should be advocated by a suitable strategy of “separating concerns” for coping with adaptive workflow management. The following diagram could demonstrate the level of abstraction for workflow adaptation concerns:



*Figure 2.1 Abstraction levels of workflow models*

Domain level concerns the business context and the business system to which the workflow system constitutes. Process level concerns the changes, including evolutionary and ad-hoc changes of workflow models and their constituent workflow tasks. The problem includes the redesign, substitution and version control of workflow models and handling the possible impacts. Resource level concerns the change in assignment of workflow supporting resources. Infrastructure level concerns how the workflow system adapts to different underlining implementation platforms and technologies.

Adaptation of workflow system to different business context often requires re-development and re-implementation of the system. Resource adaptation is another important issue, but we will not discuss in detail here. We believe that process level

adaptation would bring the most benefits to the organization in terms of performance and efficiency. After integration with other levels of adaptation, the workflow system architecture would become more comprehensive. For this reason, we discover that most work in literature focuses on process level adaptation. We will look into the details in section 2.3.

### **2.2.2. Dynamic and Flexible Process Approach**

The dynamic and flexible process approach for adaptive workflow tries to overcome the often-criticized too-rigid process structure definition of workflow management system. It allows modification to the process structure and process instance to enable them to adapt to changes in the execution environment and criteria. Such dynamic changes would give rise to difficulties in mapping the instance back to its schema and handling potential impact to other instances and referenced data.

GPSG proposed by [Glance, Pagani and Pareschi 96] view documents and tasks as duals to one another and exploits constraints to express the soft dependencies among related activities. It takes care of the process definition issue of WfMS, but does not consider administration and monitoring issues. [Chang, Gautama and Dillion 01] proposed extended activity diagrams for adaptive workflow modeling. It visualizes large and complex workflow systems. Under this modeling, work for a process is divided into actions with pre-conditions. An action is only fired when it matches the pre-conditions. The arrangement of actions is done by human decision and is not automated. [Van der Aalst 99] proposed a generic model to describe dynamic workflow and provide ways to handle the impact from changes in process structure during execution. However how to change the structure is a human decision and is not automated. [Ellis, Keddra and

Rozenberg 95] also addressed on dynamic changes within workflow systems. They used petri net to model workflow and give a set of operations to implement the dynamic changes. This piece of work is merely a representation of a dynamic system but the way that the structure changes is left to human decision and is not automated. [Bider and Khomyakov 00] proposed a state-based flexible approach to workflow management. However this approach imposes too few controls over the process flow, which may result in loss of control. It addresses the process definition issue of WfMS architecture. [Tang 99, Tang and Xing 99] proposed ad-hoc recovery on workflow management. It discussed the possible impact of ad-hoc modification to process instances under the execution environment. It addressed the process definition issue.

[Ivanovic and Budimac 99] proposed a framework that merges information flow and maintenance with the decision making process. It contributes to the generic product structure and interoperability issues of WfMS. [Joeris 97] integrates workflow and document management by an agent-based approach. It addresses the invoked application and interoperability issues of WfMS architecture. Ad hoc recovery proposed by [Tang 99, Tang and Xing 99] addresses on the process definition issue. The approach is to enable controlled flows of instances to be diverged at run time without modifying workflow schema.

[Marazakis, Papadakis and Nikolaou 98] proposed a flexible workflow management approach by dynamic reconfiguration of work sessions in a dynamic and heterogeneous run-time environment. It provides a set of APIs for the monitoring framework. However, it does not give systematical global planning to help the work to achieve its goal. The works discussed are only partly addressing the speed-up in workflow instance in literature. There are no speed-up facilities provided in products like Flowmark and

InConcept.

[Kafeza and Karlapalem 01] proposed the speed-up of workflow instance by greedy algorithm. In their work, they exploited alternative paths in the workflow model. Then advanced scheduling based on greedy algorithm was applied to the alternative paths available on the process definition to find out the way with the least cost to completion. One disadvantage for greedy algorithm is that it may be trapped in a local maximum when optimizing process performance.

[Lucertini, Pacciarelli and Pacifici 99] proposed a framework for building layout for optimal performance in material flow management in assembly systems. In their research work only a basic infrastructure is proposed. No actual method for optimization is addressed.

[Huth, Smolnik and Nastansky 01] applies topic map to ad hoc workflows in order to manage process knowledge. The accumulated knowledge improves the process structure design. It addresses the process definition issue. [List, Schiefer and Bruckner 01] studied the process warehouse approach for knowledge management in workflow process design. It helps capture the practices and norms of an organization. It addresses the process definition issue. [van der Aalst; Verbeek and Kumar 01] uses exchangeable routing language (XRL), which describe processes at an instance level, to represent process definitions. Since XRL is instance-based, workflow definitions can be changed on the fly and sent across organizational boundaries. It addresses the process definition issue. [Heinl, Horn, Jablonski, Neeb, Stein and Teschke 99] stated that there are two methods for providing flexibility to workflow management, namely selection and adaptation. Based on this idea, [Halliday, Shrivastava and Wheeler 01] proposed OPENFlow system. Flexibility by selection is achieved by ensuring that there are a number of execution paths

in the workflow process so that key decision points are well represented. Flexibility by adaptation permits dynamic changes to process structures to include one or more new paths. This work addressed the process definition issue of WFMS architecture. [Zainudin and Hamdan 01] proposed a design for workflow engine, recursive definition and execution of workflow tasks. It addresses the process definition issue. [Grigori, Charoy and Godart 01] addressed flexible data management and execution to support cooperative workflow. With the aid of Coo transaction protocol, modeling process structure according to a pre-defined manual and controlling them as in actual enactment becomes possible. It addresses the process definition issue of WFMS architecture. [Haake and Wang 97] proposed Cooperative hypermedia based process support system aimed at flexible business process support. It provides the coordination and communication support for structured to unstructured business processes. It addresses the process definition issue. [Borgida and Murata 00] provided a framework for data and process on workflow system. It enables tolerating exceptions during workflow execution. It addresses the process definition issue.

### **2.2.3.Extended Inter-organizational Approach**

The focus of research for inter-organization workflow is on data interchange rather than business process support. In fact many activities in electronic communication involve processes across multiple organizations. There are efforts on the inter-enterprise process supported by Workflow technology [WFMC 96].

XML [W3C 01] provides a facility to model complex data structure in a common and exchangeable format. It allows inter-enterprise data interchange for heterogeneous and potentially incompatible systems. Internet scale standard interoperability of workflow is

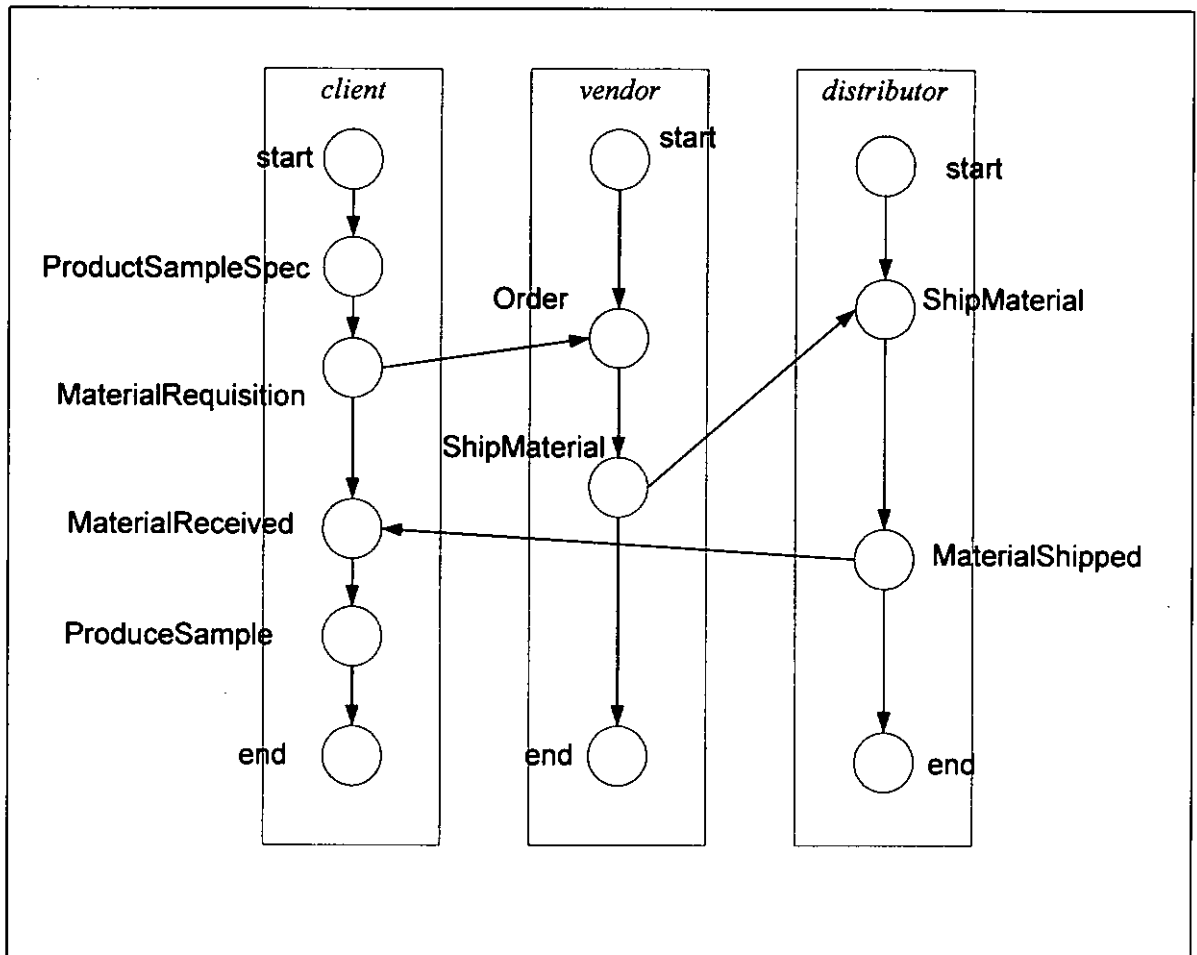
being proposed by WfMC [Hayes, Peyrovian, Sarin, Schmidt, Swenson & Weber 01]. A message set based on XML is defined. This message set is used for communication amongst agents from different organizations for workflow interoperability. In fact the work provides a descriptive infrastructure for the e-commerce environment. It does not provide facilities to speed up and increase the efficiency of the activities in the E Commerce environment. Extensible Database system for E Commerce [DeFazio, Krishnan & Srinivasan 01] is another approach for inter-operability in E Commerce. With the technology of XML, extensible storage, indexing, computation, data projection and representation are implemented on the Oracle 8i: DBMS platform. This project provides only a descriptive infrastructure for the inter-organizational environment but not an optimization and speed-up mechanism. XML is actually a declarative approach to enterprise integration. It does not concern performance optimization of logic on activity execution.

Component-based Software Development is used to provide a 3-tier Web-based enterprise application in [Pour 99]. The aim of this approach is to provide a timely manner to develop and improve reliability, maintainability and overall quality. The benefit on performance of this approach is limited to software design only. Activity semantics improvement cannot be expected.

Protocol for E-commerce [Xing, Wan, Rustogi & Singh 01, Papa, Bremer, Hale & Shenoi 01] provides an infrastructure for inter-enterprise communication and interpretability. Sophisticated e-commerce applications should be further implemented based on this primitive communication channel.

[Van der Aalst 99] proposed a process-oriented architecture for E-Commerce and inter-organizational workflow. It considered the inter-organizational activities with two

forms of interoperability: case transfer and loosely coupled. In the form of case transfer, each business party owns a copy of the workflow process definition. Cases are actually instances of the run-down of the process. Cases may be transferred from one party to another. The following figure shows a case-based inter-organizational workflow:



*Figure 2.2 Inter-organizational workflow process*

Some local variation may exist and mapping of the states under different variations is defined to allow proper transfer of cases. It does not concern performance optimization of logic on activity execution.

[Gal & Montesi 99] proposed a model for inter-enterprise workflow management systems. The model proposed attempts to extend the standard workflow to enable the description and control of the full life cycle of activities in supply chain management. The objective

of the model is to ensure collaboration, where all parts of the system work together for achieving the same goal, and submission, where a central mechanism provides instructions to be followed by the various parts of the system for a supply chain. Under this model a common workflow model is extended to have points of entry and exit to allow inter-enterprise communication. Workflow virtual machines are implemented in each organization in order to allow storage of local variables and perform inter-enterprise with each other based on these variables. However, this model does not consider how to arrange the connection between entry points and exit points to give better performance.

Crossflow project [Hoffner, Ludwig & Culcu 00] attempts to support the dynamic establishment and enactment of a business relationship between two organizations. It is further enhanced by automating the set up of the contract enactment and supervision infrastructure, and by connecting them together to allow the business process of the partners to cross their organization boundaries. Crossflow uses a business end-to-end approach to provide support to a business process cycle for the inter-organizational environment, which includes service contract establishment, enactment, termination and analysis. However, the main contribution of Crossflow lies in inter-enterprise operability, and not in the idea of optimization of activity arrangement for better performance.

[Zeng, Ngu, Bentallah & O'Dell 01] build an agent based workflow architecture supporting inter-organizational workflow using their component-based workflow model. In this model, tasks may be dynamically assigned to service providers during the enactment of a cross-enterprise workflow. However, the arrangement of tasks is not optimized to give best efficiency.

XPECT [Andreoli, Pacull & Pareschi 00] uses Coordination Language Facility (CLF) to coordinate distributed objects in a framework for E Commerce. It does not provide

facilities to speed up and increase the efficiency of the activities in the E Commerce environment.

[Ling and Loke 01] uses a mobile agent to enable inter-organizational workflow. An itinerary algebra is developed to ensure the liveness of each agent, which is essential for providing correct and reliable collaboration among all parties under the mobile environment. It addresses the inter-operability issue. [Le Pallee and Vantroys 01] proposed meta-object facility and inheritance to improve cooperation in WFMS and hence to support inter-organizational workflow.

#### **2.2.4. Other Workflow Management Approaches**

As workflow management system collaborates work of different members in organization, there would arise an issue of concurrency control, which means handling the coordination of workings by different concurrent parties and resolving potential conflicts. [Lee, Han and Shim 01] analyze possible read-write and write-write conflicts for concurrent workflow definitions. This is also important for WFMS inter-operability.

[Lee, Han and Lee 01] address how to handle the wireless environment with unreliable connection when deploying workflow management application to such platform. With the support of disconnected (offline) operation, adaptability to the mobile environment is provided to WFMS. It addresses the inter-operability issue of WFMS architecture.

Knowledge management for workflow is studied in [List, Schiefer and Bruckner 01]. It uses the process warehouse approach for knowledge management in workflow process design. It helps capture the practices and norms of an organization. It addresses the process definition issue.

## 2.3. Concluding Remarks

The contribution of the researches discussed regarding WfMS architecture issues can be summarized in the following table:

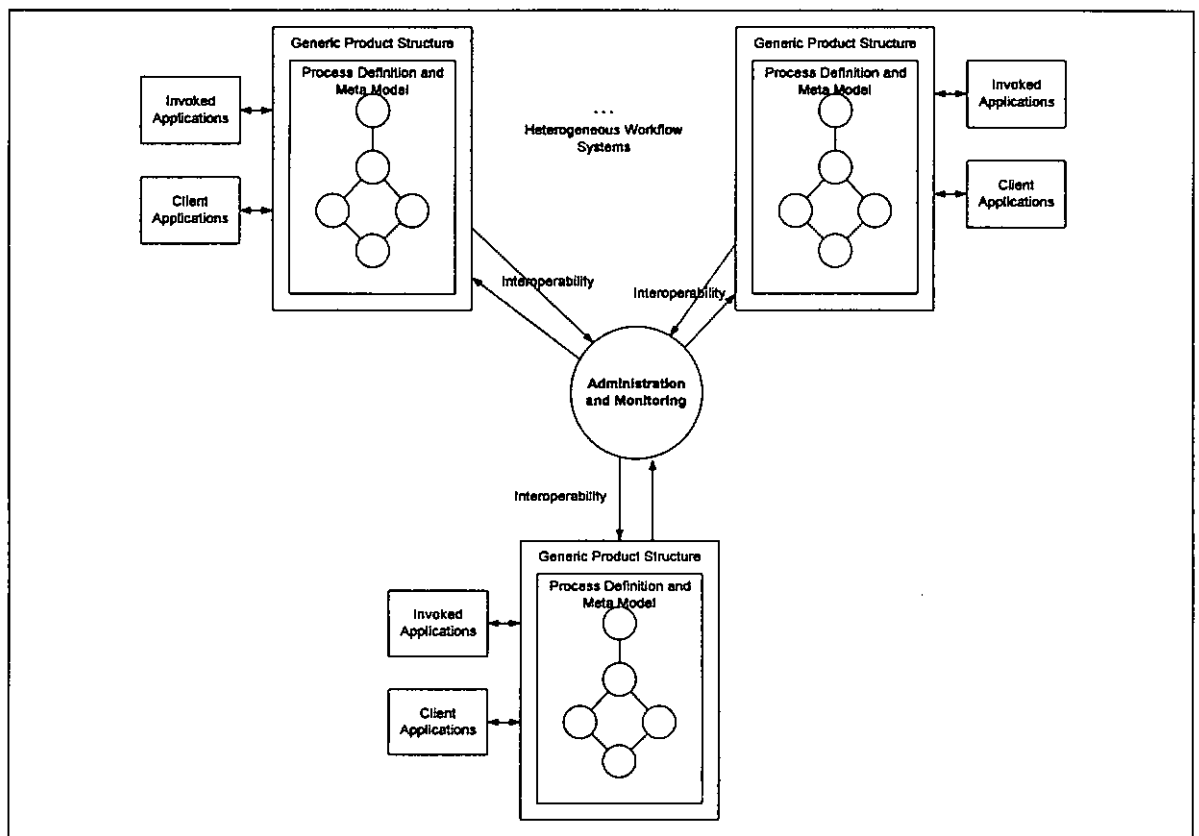
*Table 2.1 Issues covered by literature in adaptive workflow management*

Research work/Package	Generic Product Structure	Process Definition	Process Definition Meta Model	Client Application	Invoked applications	Interoperability	Administration and monitoring
[Glance, Pagani and Pareschi 96]		✓					
[Chang, Gautama and Dillion 01]		✓					
[Aalst 99]		✓					
[Ellis and Rozenberg 95]		✓					
[Bider and Khomyakov 00]		✓					
[Tang 99, Tang and Xing 99]		✓					
[Ivanovic and Budimac 99]	✓	✓				✓	
[Joeris 97]					✓	✓	
[Tang 99, Tang and Xing 99]		✓					
[Marazakis, Papadakis and Nikolaou 98]		✓					
[Kafeza and Karlapalem 01]		✓					
[Lucertini, Pacciarelli and Pacifici 99]		✓					
[Huth, Smolnik and Nastansky 01]		✓					
[List, Schiefer and Bruckner 01]		✓					
[van der Aalst, Verbeek and Kumar 01]		✓					
[Halliday, Shrivastava and Wheeler 01]		✓					
[Zainudin and Hamdan 01]		✓					
[Grigori, Charoy and Godart 01]		✓					
[Hayes, Peyrovian, Sarin, Schmidt, Swenson & Weber 01]						✓	
[Pour 99]					✓	✓	
[DeFazio, Krishnan & Srinivasan 01]					✓	✓	
[Xing, Wan, Rustogi & Singh 2001, Papa, Bremer, Hale & Sheno 01]						✓	
[Aalst 99]						✓	
[Gai & Montesi 99]					✓	✓	
[Hoffner, Ludwig & Culcu 00]						✓	
[Zeng, Ngu, Bentallah & O'Dell 01]						✓	
[Andreoli, Pacull & Pareschi 00]						✓	
[Ling and Loke 01]						✓	
[Le Palle and Vantroys 01]						✓	
[Lee, Han and Shim 01]						✓	
[Lee, Han and Lee 01]							
[List, Schiefer and Bruckner 01]							

From the review in literature we see that the current research effort on adaptive workflow focuses on dynamic feature modeling and on the capability to handle the impacts from

dynamic changes in process structure. These approaches address the inter-operability issue of workflow system architecture. However, few or even none of them concerns automated performance optimization and evaluation of current practices, which should address the administration and monitoring issues. Current adaptive workflow research work is weak in this area despite that it plays an important role in a comprehensive workflow management solution. We may check this fact out in the following diagram.

Work done on workflow adaptation and exceptional handling already provides a good and robust but rather primitive basis for adaptive workflow management. We believe the right direction for the delivery of a comprehensive solution for adaptive workflow management is to build a framework for optimizing process structure and monitoring performance of process instance.



*Figure 2.3 Integration of proposed framework*

# 3. Overview of Genetic Algorithm

## 3.1. Preliminaries

### 3.1.1. The method

In the 1970s, Holland invented genetic algorithm as a computer program that mimics the evolution process observed in the natural world. Such evolution has the following features:

- Evolution operates on the encoding chromosome rather than on the object the chromosome encodes for.
- The chromosome and the performance of the decoded structure are linked by natural selection. Such linkage allows chromosome that encodes more successful structures to reproduce more often.
- Evolution takes place when reproduction is performed. Such evolution includes recombination of features from parents to form new children. Mutation may cause such children to have some features not possessed by their parents.

Genetic algorithm has the following elements:

*Population as search space to optimize*

The algorithm manipulates a population, which is a set of candidates for possible solutions to a problem. This forms the search space to an optimization problem.

*Chromosome*

Chromosome is the encoded representation of a solution to a problem. There are many ways to encode a solution to a chromosome. More discussion would be on section 3.2.

### *Genetic Operation*

Genetic operations include selection, crossover and mutation. More discussion would be on section 3.2.

### *Each encoded chromosome is associated with fitness*

Each chromosome is associated with a fitness that reflects how good it is in solving a problem.

A simple genetic algorithm can be demonstrated as follows:

### The genetic algorithm

1. Initialize a population of chromosomes
2. Evaluate the fitness of each chromosome in the population
3. Reproduce new chromosomes by mating current chromosomes. Mutation may be applied as parents mate
4. Delete members in population to make room for new chromosomes produced.
5. Evaluate the new chromosome and insert it into the population
6. If time is up, terminate the iteration and return the best chromosome as solution. Otherwise go back to step 3.

*Figure 3.1 The genetic algorithm*

### 3.1.2.Encoding

Fundamental to the GA Structure, encoding is the process of representing competitive features of a solution, known as phenotype, in a chromosome, namely the genotype of the solution. The detailed way to encode solutions depends on the nature of the problem.

Usually chromosomes are in the form of a string of digits or binaries. Examples:

```
00000110011000010111  
01689400665066
```

A large number of optimization problems have real-valued continuous variables. A common way to encode them is to use their integer representations. Each variable is first linearly mapped to an integer defined in a specified range, and the integer is encoded

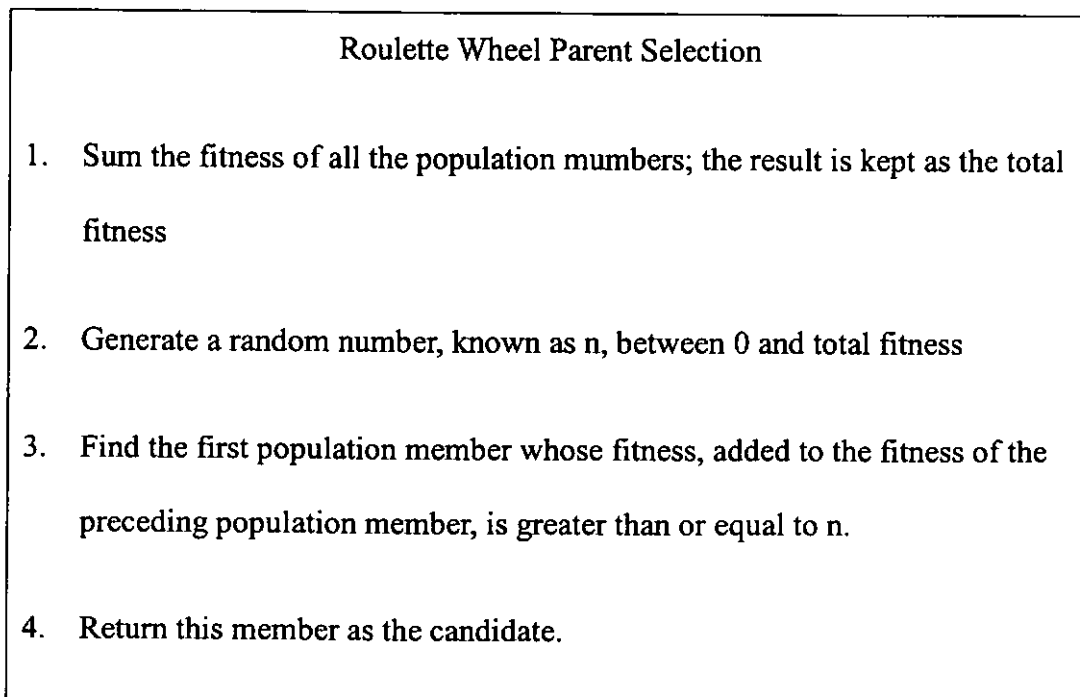
using a fixed number of binary bits. The binary codes of all the variables are then concatenated to obtain a binary string.

### **3.1.3. Genetic Operations**

Operations in Genetic Algorithm include parent selection, crossover and mutation. Each operation has an occurring probability.

#### **3.1.3.1. Roulette Wheel Parent Selection**

In a natural evolution process, the fitter the population member, and the more the chance it reproduces. Roulette wheel selection is a commonly used technique that tries to mimic this fact.



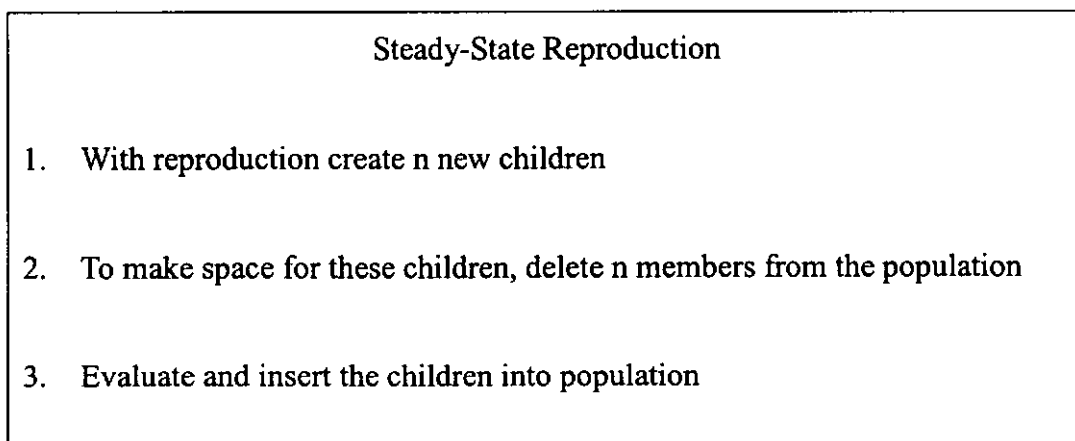
*Figure 3.2 Parent Selections*

### 3.1.3.2. One-point Crossover and Mutate

Crossover swaps randomly selected sections of two parents to produce two new children, which are genetically different from their parents and are, hopefully, better than their parents in fitness. Crossover is an extremely important component of genetic algorithm, making it different from other optimization methods such as those using empirical tests. If such operation is removed from the algorithm, the performance will be much lower.

### 3.1.4. Steady State GA

When new children are reproduced, they will replace existing members in the population. To avoid the problem of some good members in the population not reproducing at all and the good feature in the population being lost in crossover and mutation, a reproduction technique must be employed. Steady-state GA replaces only one or two individuals at a time rather than all the individuals in the population. It can prevent the above problem.



*Figure 3.3 Steady-State Reproductions*

## 3.2. Genetic Algorithm and Graph problem

In the topology optimization problem proposed in [Schiffmann 00], a generic framework for encoding in the evolution algorithm is proposed. The method is a combination of direct encoding and parametric encoding. Direct encoding does not use a separate

genotype representation, but encodes the phenotype directly. This is done by representing the connection in a matrix. Parametric representation uses numeric values such as the number of hidden layers, the number of hidden nodes and the connection pattern among different layers to describe the network. The central idea is to interpret the nodes in the network graph as layers that are themselves further specified by parametric representations.

Genetic algorithm has been proposed to solve optimization of graphs. [Barreto and Barbosa 01] layout graphs using genetic algorithm. Their algorithm tries to minimize several aesthetic criteria. In the graph-based approach for sorting network problem in [Choi and Moon 01], each sorting network is represented by a chromosome. The corresponding pair of input buses of comparator for sorting networks represents each gene. Evolutionary approach for solving degree-constrained minimum spanning tree problem has been proposed in [Knowles and Corne 00]. The genetic algorithm used employs the randomized primal method (RPM). [Magyar, Johnsson and Nevalainen 00] uses a hybrid genetic algorithm to solve the 3-matching (3MP) problem, and NP-complete graph problem. It uses integer strings to represent the connected nodes in the graph. Genetic algorithm is also used to optimize data flow graph with multiple objectives [Landwehr 99]. Under this approach, genes represent the sub-expression, and gene forms the chromosome that represents the whole expression. Fuzzy-graph based genetic algorithm is proposed in [Qian 99] with application to image interpretation. [Ali, Nakao, Tan and Chen 99] presents an evolutionary algorithm for graph coloring problem. Colors and edges are encoded in a 2-dimensional artificial string. As part of the routing problem of all-optical network, the power assignment problem is solved by genetic algorithm in [Ali, Ramamurthy and Deogun 99]. In [Tang, Eshraghian and Cheung 99]

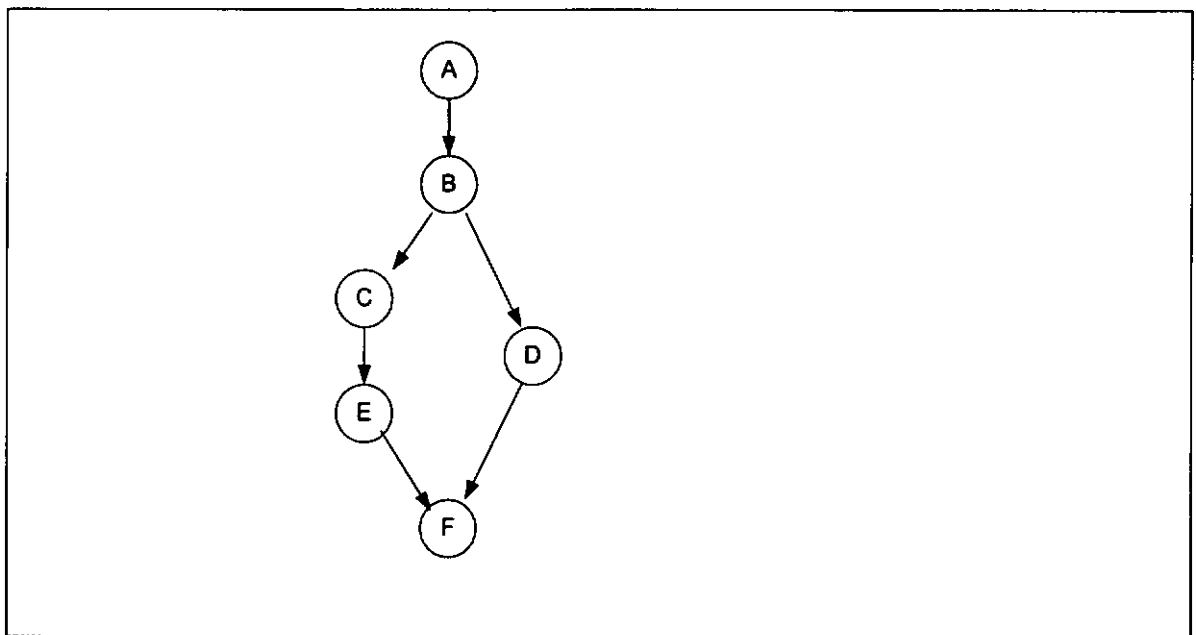
genetic algorithm is used to perform constrained via minimization. Via is a mechanism (hole) through which wire segments that belong to the same net but distributed on different layers are inter-connected. This is a typical optimization problem in very large scale integrated circuit (VLSI) routing. In the proposed algorithm, a chromosome is a switching plan in the form of a binary 2-dimensional matrix that denotes connections of vertices. In [Chen, Fu and Lin 00], a genetic algorithm embedded search strategy over a hybrid color-timed Petri-Net for wafer fabrication is proposed. Indirect approach is used in chromosome representation to phenotypes, which describes the features that set up the routing. [Tabbara, Dana and Mansour 00] addresses the problem of decomposing a weighted graph into a specified number of sub-graphs such that these graphs have balanced sums of vertex weights and minimal sums of edge weights. In the proposed hybrid genetic algorithm, vertex of the graph solutions are arranged in chromosomes, the arrangement of which represents how they are connected. It is a direct approach for encoding. Dynamic rule graph plays an important role in data mining applications. The drawing process involves dynamic models. Genetic algorithm is proposed to solve the problem in [Kuntz, Lehn and Briand 00]. In the proposed algorithm, chromosomes represent the arcs in dynamic rule graph directly.

Design of optimal network is an optimization problem. Genetic algorithm has been proposed to solve such problem. [Sawionek, Wojciechowski and Arabas 99] One interesting part of the problem is how to encode graph design into chromosome string for genetic algorithm. This is a special genetic algorithm encoding method.

### **3.3. Proposed Graph Optimization by Genetic Algorithm**

Our proposed work tries to find an optimal arrangement of tasks in a process structure with given constraints. This problem can be solved by the graphical approach. How the

workflow problem being modeled by graph will be discussed more in detail in section 4. A graph actually can be modeled with a set of nodes and edges,  $G = \{N, E\}$ . How a graph differs from another lies in its different connection by edges between nodes. Hence we may denote a connection in a graph by  $C = \{N1, N2\}$ . The graph design can be described by all connections between the nodes. A 2-dimensional matrix may represent such connections. In this matrix, each 1 in the matrix means there is a connection between the nodes denoted by the row to the node denoted by the column.

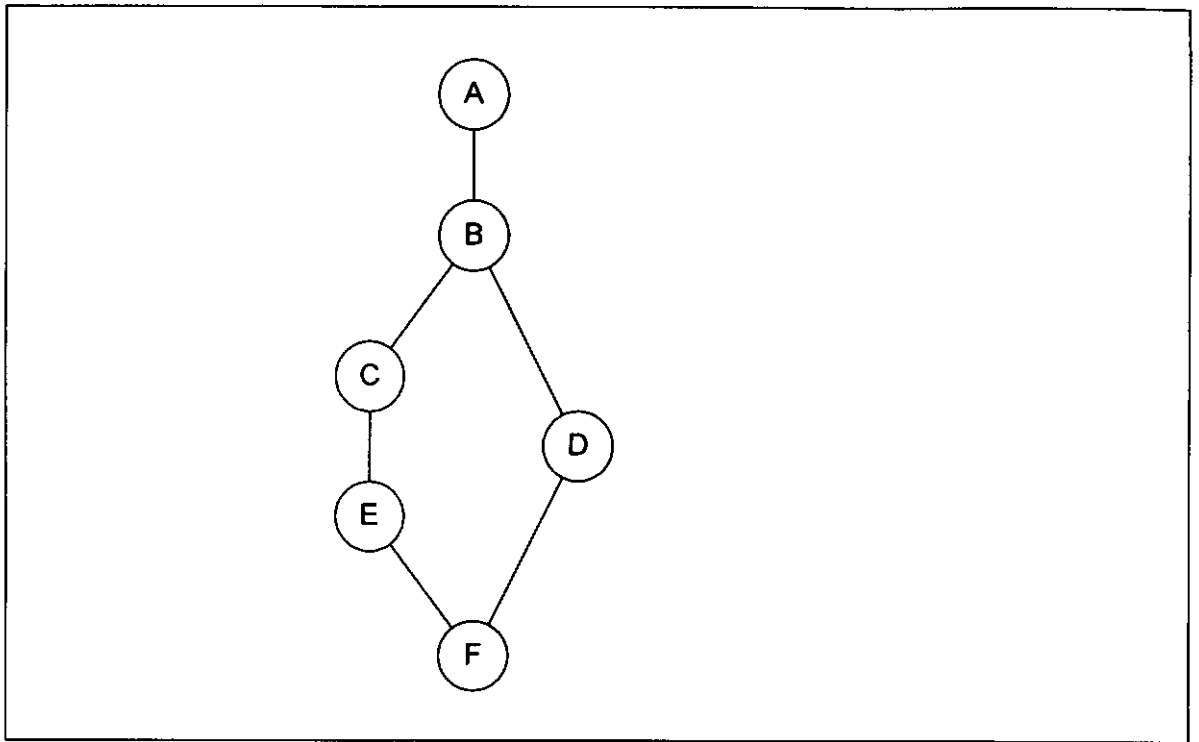


*Figure 3.4 Example of process as connection of nodes*

	A	B	C	D	E	F
A						
B	1					
C		1				
D		1				
E			1			
F				1	1	

*Figure 3.5 Corresponding connection matrix representation for the process*

The above representation denotes a directed graph. An undirected graph may be denoted by the union of the matrix with its transpose.



*Figure 3.6 Example of undirected graph*

	A	B	C	D	E	F
A		1				
B	1		1	1		
C		1			1	
D		1				1
E			1			1
F				1	1	

*Figure 3.7 Connection matrix for an undirected graph*

### 3.4. Difference between Job-shop scheduling problem and proposed Workflow Process optimization problem

Job shop scheduling is a problem of finding the optimal allocation of jobs to multiple processors running in parallel [Fox 94]. This problem has some similarity to our process optimization problem. However, it cannot handle many issues that our proposed solution had considered. We should like to have a comparison of the two problems. The similarity and difference between them is listed below:

*Table 3.1 Similarity among job shop scheduling and process optimization*

Similarity
Problem nature is that given a set of tasks or jobs, find the optimal arrangement
Supports sequencing of parallel jobs/tasks. Resultant sequence complies with given constraints.

*Table 3.2 Differences among job shop scheduling and process optimization*

Differences	
Job shop scheduling	Process optimization
Jobs are independent, use only time sequence to indicated precedence relationship	Tasks are connected with precedence relationship
Number of machines usually predefined. Hence limited concurrent jobs.	No explicit limit on concurrent tasks.

In job shop scheduling problem, although there were multiple machines to run jobs concurrently, we cannot trace out the relationship between jobs in different machine. Although the time sequence indicates their precedence relationship, the information flow was suppressed under the problem model. In process optimization, the task precedence relationship is explicitly defined and emphasized. The difference shows that our work provides a unique contribution to workflow process optimization.

### **3.5. Remarks**

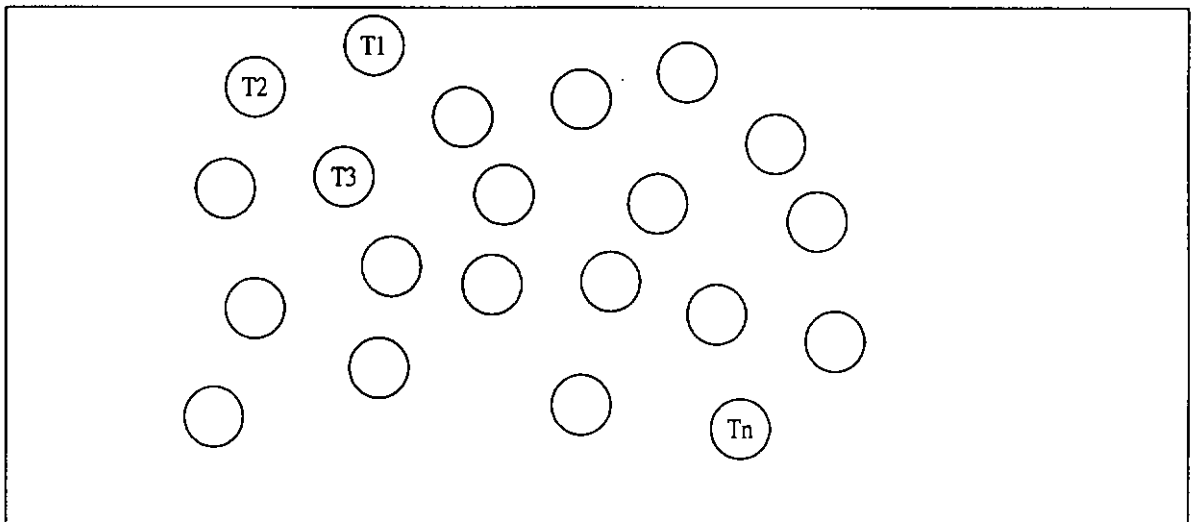
The reason for us to choose genetic algorithm to be our solution to the workflow process problem is that genetic algorithm has the ability to escape from local maximum in search space and reach the global maximum. We believe that in our problem domain, some similar process structure will have better performance and group together in the searching space to form local maximums. As genetic algorithm has such characteristics, we believe that it will be a good candidate to solve the described problem.

We have seen the traditional genetic algorithm and its variants. We have also seen how genetic algorithm is modified to solve graph optimization problems. Since a workflow process may be modeled by a graph, we can adopt such approach for our adaptive workflow framework.

# 4. Adaptive workflow approach – inter-organizational Evolutionary

## 4.1. Overview

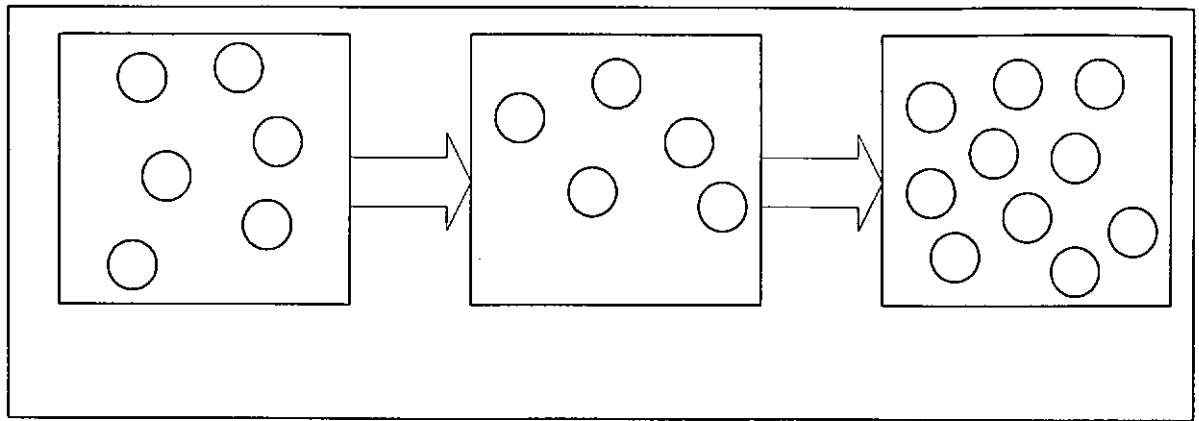
We first have an overview on what the algorithm does. Given a set of tasks for a process, we try to find an optimal arrangement for a process structure.



*Figure 4.1 Tasks before optimization*

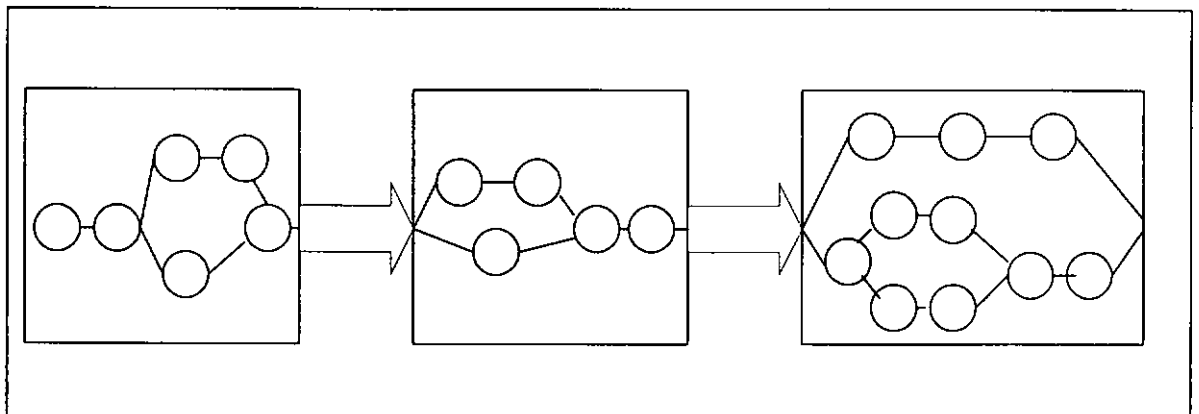
The optimization process can be divided into three stages:

Segmentation - At this stage, we partition the task in the set into segments according to organizational boundary.



*Figure 4.2. Segmentation*

Optimization of each segment - At this stage, we optimize the task arrangement inside each segment by evolutionary algorithm.



*Figure 4.3 Recombine after optimization*

Re-combine - At this stage, we recombine the segments into a whole process structure for the overall inter-organizational activities.

## **4.2. Segmentation of inter-organizational process according to organizational boundary**

Two problems arise if we find the solution by optimizing all the tasks together to yield one large process structure. First the performance of such optimization would become so slow that it makes the framework infeasible. Second, such optimization will produce execution paths that go across organizations many times, which does not agree with the

semantic of inter-organizational operations. For this reason, we have to partition the task before we perform the optimization to yield the resultant process structure.

The idea of segmenting the process is to partition the unstructured tasks of the whole inter-organizational process. Then we optimize each partition separately to form sub-section of process. Finally we recombine these sub-sections to yield the whole process as final solution.

There are many possible ways to perform the segmentation, i.e. define the boundary. We may divide the process into partitions with the same number of tasks. We choose the way of dividing the tasks of the process according to organization boundary, i.e. partition by which organization performs the tasks. In this way, the optimization process aligns with the semantics in the inter-organizational environment

#### **4.2.1. Definition of segmentation**

Given a set of tasks for an inter-organization process, we define sets for each organization's segments. We assign membership of each set to each task in the process. Each segment may be considered as a sub-process and is to be executed sequentially. For this reason, we could ensure that all tasks in a former segment are executed before those in later segments.

#### **4.2.2. Requirement of segmentation**

The information dependency requirement should be fulfilled after segmenting the tasks. To achieve this goal, we should not schedule any tasks before the tasks on which they depend in the resultant process structure. For this reason, in the set of segments produced in segmentation, former segments should not contain any task that depends on tasks in

later segments in the structure.

## 4.3. Evolutionary Algorithm for Workflow Optimization

### 4.3.1. Overall algorithm

As stated in section 3.3, a process can be modeled by a directed graph, and a graph may be encoded into genetic chromosome as a 2-dimensional array. We optimize such encoded chromosome with the process of initialization of population, selection, reproduction, fitness evaluation and, finally, the extraction of the optimal result. The overall algorithm can be described by the pseudo code in figure 4.1.

Initialize Population
Selection
Reproduction
Crossover
Mutation
Fitness Evaluation
Repeat from Selection until Predefined Number of Loops

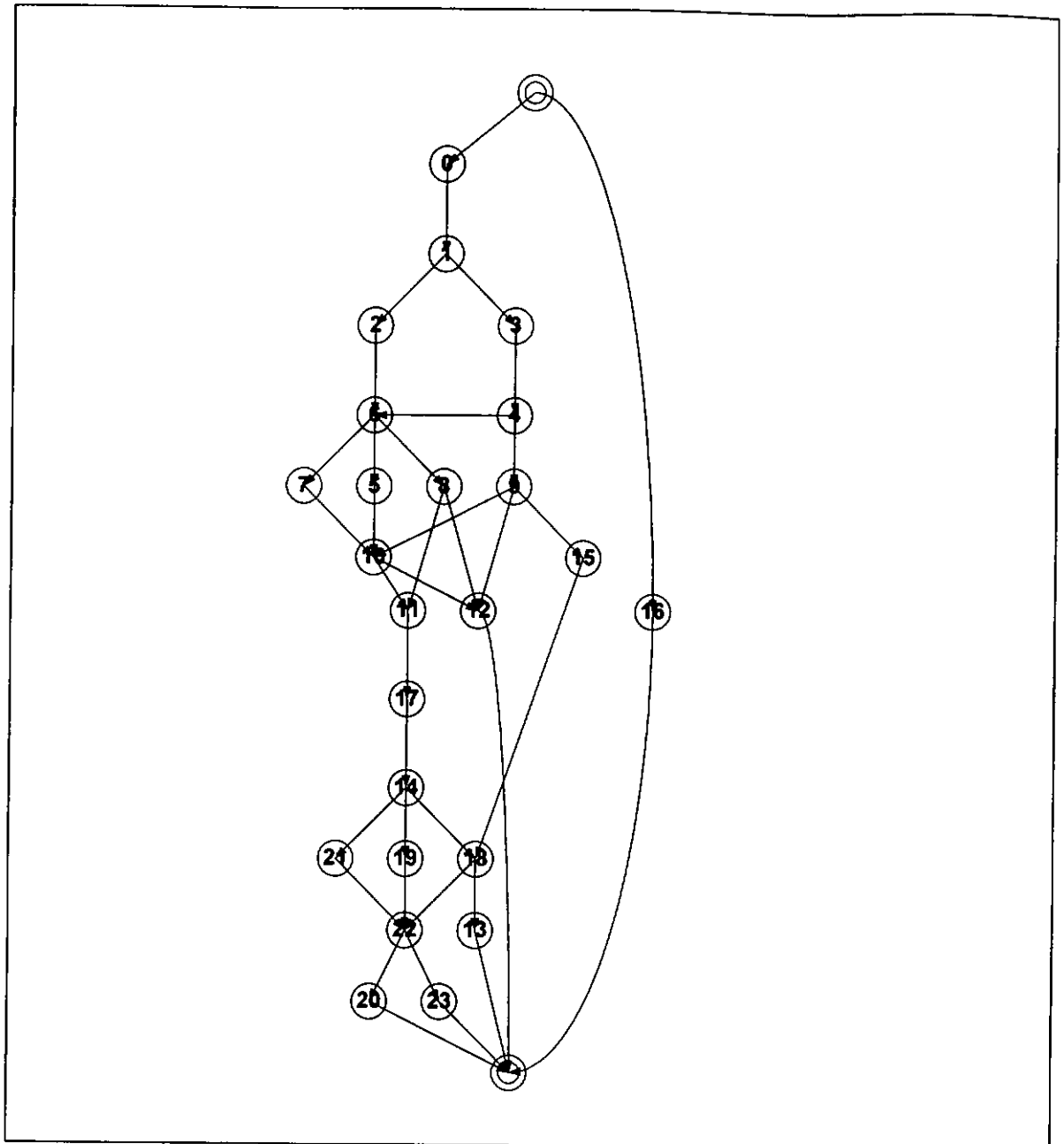
*Figure 4.3. Pseudo code for genetic algorithm*

We will give detailed explanation of the modeling approach and the implementation of genetic operations in the coming sections.

### 4.3.2. Model of a process

#### 4.3.2.1. Assumption

We assume that each workflow process consists of connected tasks. The connection between each task is directional, indicating the process flow. Each process has one or more startup and finishing tasks. For this reason, a workflow process can be represented by a connected graph, as shown below:



*Figure 4.4 A Process example*

#### **4.3.2.2. Process and precedence relationship of tasks**

We had defined the formal representation of the problem in section 1.4. In our case each process is actually a directed graph with nodes as tasks and edges as transitions from one task to another. [Sawionek, Wojciehowski & Arabas 99] provided a solution for optimizing graph design with evolution algorithm. As stated in section 1.4, we denote

each process in this form,  $P = (T, E, R)$ , where  $T$  is a  $p$  element set of nodes representing tasks in the process and  $E$  is a  $q$  element set of edges representing transitions from one task to another after it has been completed. A connection matrix for  $P$  is a  $p \times p$  matrix with the entry  $a_{ij}$  equal 1 if there is a connection from the  $i^{\text{th}}$  node to  $j^{\text{th}}$  node, and 0 otherwise ( $i, j = 0, 1, \dots, p-1$ ).

#### 4.3.2.3. Constraints and feed-backward information

As one task is completed, information from it will be used by the next task. For each task, there is a precedence relationship with other tasks. We have denoted such relationship in section 1.4 as  $Pre(T_a, T_b)$  where  $T_a, T_b \in T$ . Information dependency constraints specify that for certain tasks they must contain information from some other tasks. Such constraints could be regarded as a set of rules given by  $R(T_a, T_b)$ , where  $T_a, T_b \in T$ . A process structure  $P$  is said to satisfy the constraints set  $D$  if the following condition holds:

$$Pre \supseteq R$$

The precedence relationship amongst tasks is the basis of fulfillment of constraints to the workflow structure. As some tasks depend on information or results from other tasks, they cannot be executed before the tasks they depends on. The following examples show some constraints and how they affect the workflow process structure. A process consists of six tasks, namely Task A to F. Some constraints must be fulfilled for this set of tasks:

Constraint 0: All tasks must get information from Task A

Constraint 1: Task C must get information from Task B

Constraint 2: Task D must get information from Task B

Constraint 3: Task F must get information from all other tasks

The above constraints give the following set of constraint rules:

$$\begin{aligned}
 R &= \{R_1, \dots, R_{11}\} \\
 R_1 &= R(A, B), R_2 = R(A, C), R_3 = R(B, C), \\
 R_4 &= R(A, D), R_5 = R(B, D), R_6 = R(A, E), \\
 R_7 &= R(A, F), R_8 = R(B, F), R_9 = R(C, F), \\
 R_{10} &= R(D, F), R_{11} = R(E, F)
 \end{aligned}$$

Figure 4.5 Constraint rules

Possible valid processes that fulfill these constraints include:

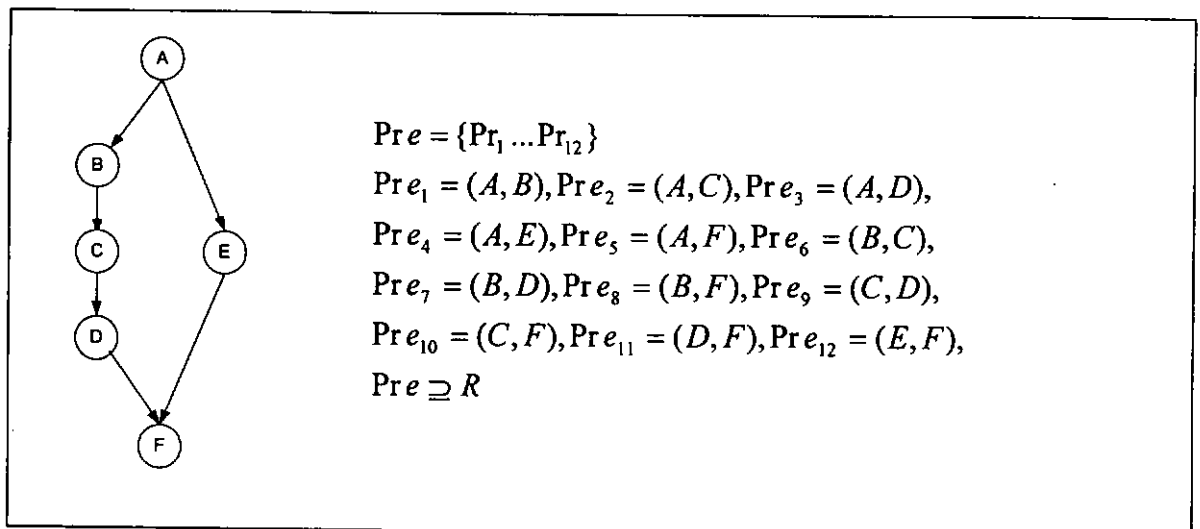


Figure 4.6 Information carriages of tasks

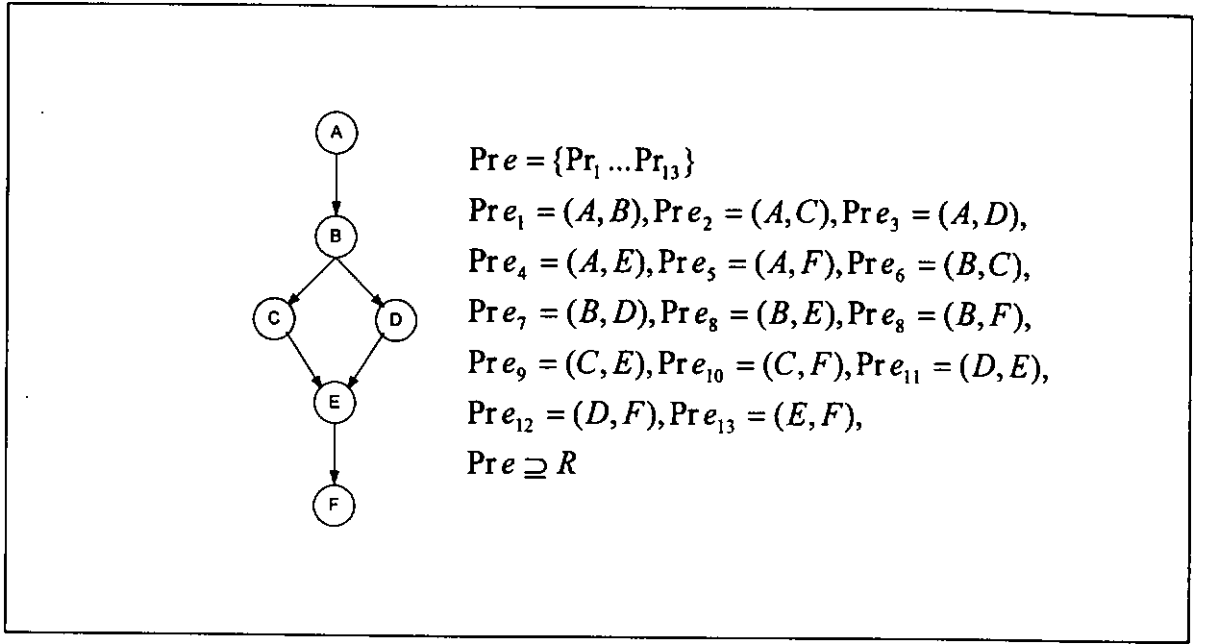


Figure 4.7 Information carriage of another process

Up till now we have assumed that all information dependencies are feed-forward. Actually there may exist some dependency that is feed-backwards. This means that any tasks between this feed back pair may need to be re-executed since information from the tasks they depend on may be changed due to backward feeding. We distinguish such information dependency from normal ones since such dependency will incur repeated execution of certain tasks.

We used the following representation for such feed-backwards in section 1.4:

$$F = \{F_1 \dots F_l\}$$

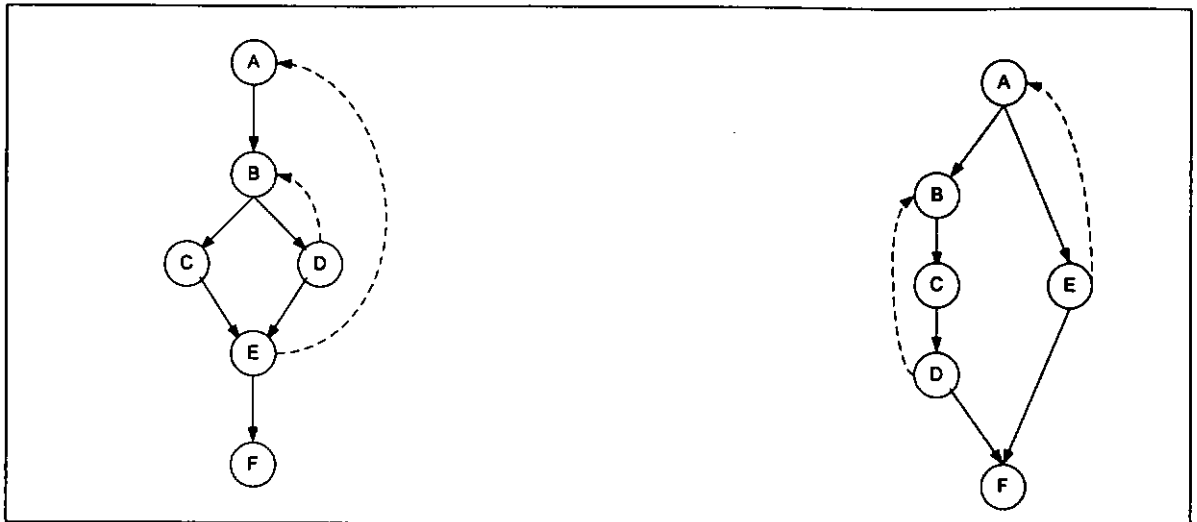
$$\forall x \in \{1 \dots m\}, F_x \Rightarrow P(T_c, T_d) \text{ and } \bar{\ell}_b(T_c, T_d), \text{ where } c, d \in \{1 \dots n\}$$

Some examples of feed-backward information are as follows.

Feed Back 0: Task D may feed back to Task B

Feed back 0: Task E may feed back to Task A

#### 4.3.2.4. Cost incurred by feed-backward



*Figure 4.8 Feed-backward*

As suggested by the term feed-backward, such kind of dependency implies changes have been made to work done previously. The repeated execution of previous tasks incurs additional costs. Such cost can be measured by evaluating the amount of tasks to be repeated.

How to count the amount to be repeated if a feed-backward occurs? To solve this problem we have to evaluate the travelling distance from the destination to the origin of feed-backward. However some parallel paths in such travel may exist. Since their effects go in parallel, only the longer path, which incurs the larger cost, would be significant to the overall performance. For this reason, we define the cost for a feed-backward as the length of the longest path for traveling from the destination to the origin of the feed-backward.

### 4.3.3. Chromosome representation of workflow process

#### 4.3.3.1. 2-dimensional matrix for task connection

[Sawionek, Wojciechowski & Arabas 99] used a two dimensional matrix to represent the connections of nodes of a directed graph. Each row and column of the matrix represents a node. If there exists a connection from node A to node B, a "1" is put into the intersection of the row and column of the two nodes. This is suitable for representing workflow, which is a set of connected tasks as we assume. Each column and row represents a task. Each "1" element in the matrix indicates there is a connection from the task of the row to the task of the column.

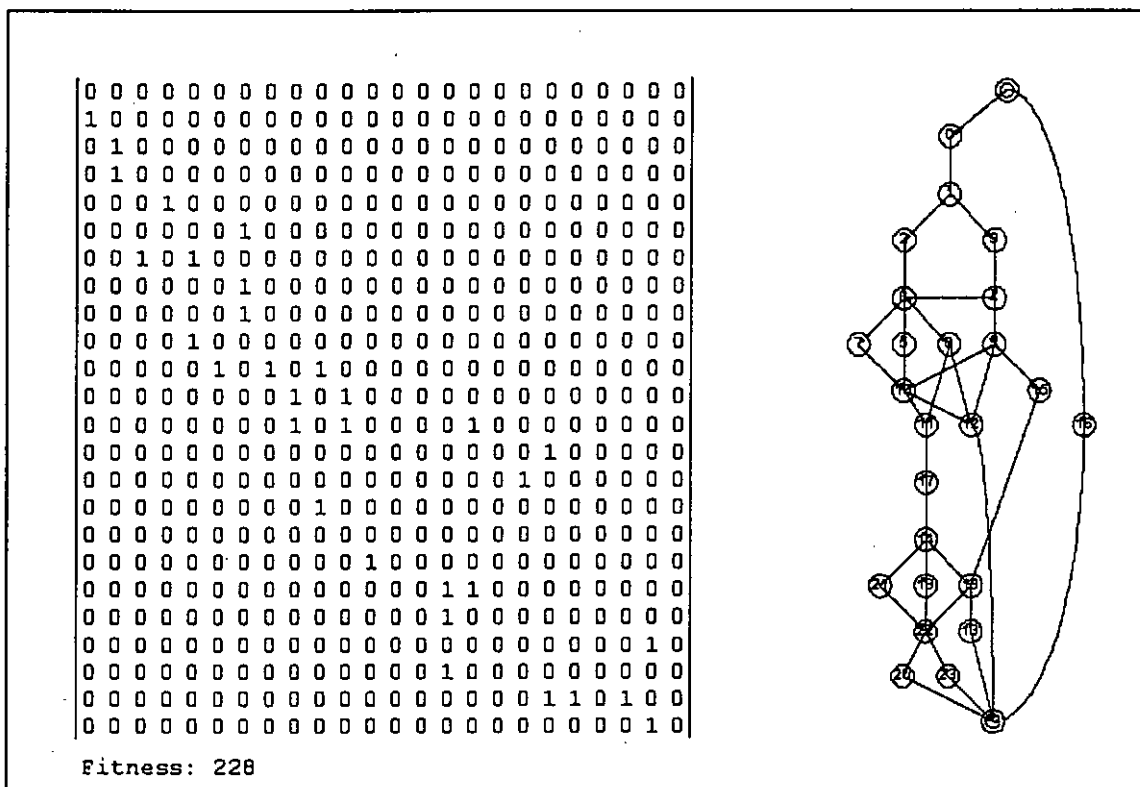


Figure 4.9 Connection matrix for the example

#### 4.3.4. Initial Population Generation

To start the evolutionary algorithm, we need an initial population set of solutions, which

is usually generated randomly. The connections among tasks should be randomly established in order to generate random solutions. However, since the solutions should fulfill the information dependency constraints, such connections could not be arbitrarily established. For this reason, the “randomness” should be constrained. As specified in information dependency constraints, each task should have preceding and succeeding tasks.

We use the following approach. For each task, we randomly select tasks from its preceding and succeeding tasks in the constraint specification, instead of adopting the whole set of tasks in the process. Then we assign connections from the selected preceding task to the task, and from it to the succeeding tasks. In case the same connection already exists, we do not add the connection in order to avoid too many redundant paths among the process structure.

The following steps can illustrate the detailed algorithm design for population generation:

*Step 1: Randomly reorder the tasks, put into a list. Process the task in random order*

To provide the randomness to the generated results, we process the task in random order.

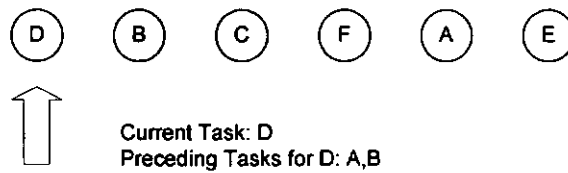
This would provide more possibility in population generation.



*Figure 4.10 Tasks before optimization*

*Step 2: For each Task, identify all tasks that should be as its preceding task as specified in the constraints.*

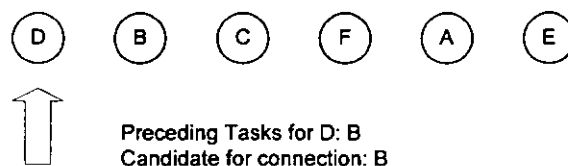
We start from the beginning of the randomly ordered list and process each task. For each task in process, we mark it as the current task.



*Figure 4.11 Information dependency for Task D*

*Step 3: For each preceding task, identify all tasks that contain information from it.*

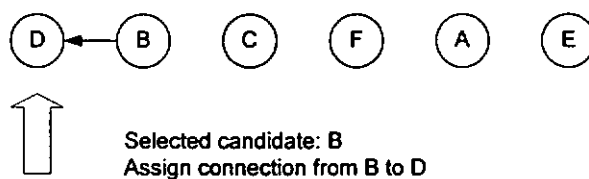
Most apparently, the task that contains information from the preceding task is the preceding task itself. However, this task may have connection to other tasks, which may also contain information from it. For this reason, it is equivalent for the current task to assign connection from the preceding task or its successors. The set of task that contains information from the preceding task, which includes the preceding task itself, and all its successors, forms the candidate set for assigning connection to the current task.



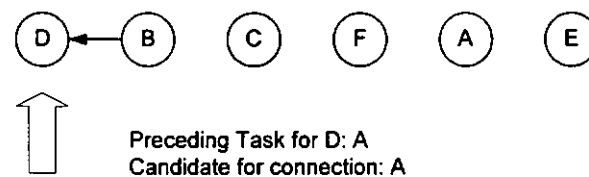
*Figure 4.12 Candidates for proceeding task of Task D*

*Step 4: Randomly select one from these tasks, assign connection from it to current task.*

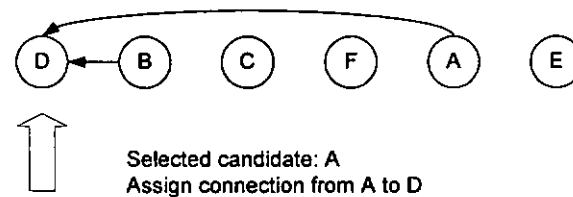
As one connection is enough, we randomly select a task from the candidates and assign connection for selected candidate to the current task.



*Figure 4.13 Assignment of connection from Task B to Task D*



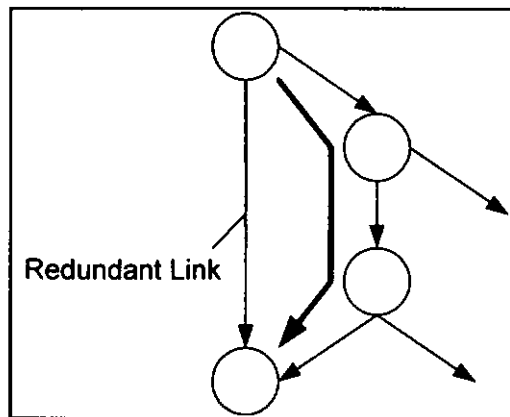
*Figure 4.14 Information dependency for Task D*



*Figure 4.15 Assignment of connection from Task A to Task D*

*Step 5: Removal of Redundant link*

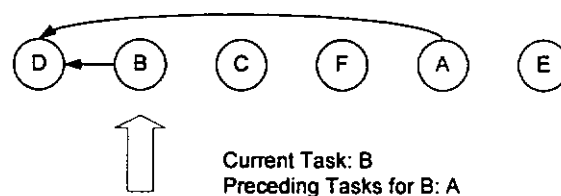
Assigning connection in such way may produce many redundant connections, which are in the following form:



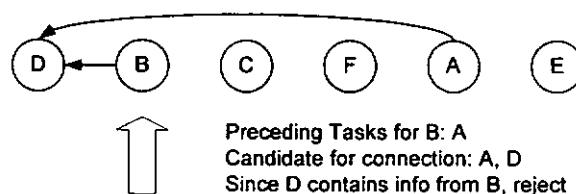
*Figure 4.16 Redundant Link Example*

In the above example, the direct link we pointed out actually cannot be utilized since an alternative with more transition of nodes exists. It cannot be executed before the longer alternative finishes its execution. We have to remove these links to make the resultant process structure more sensible. Detection and removal of such connection must be performed each time we add new connections to the graph.

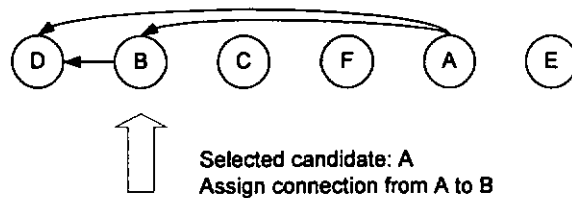
When we advance the algorithm to Task B, the above situation may occur:



*Figure 4.17 Information dependency for Task B*

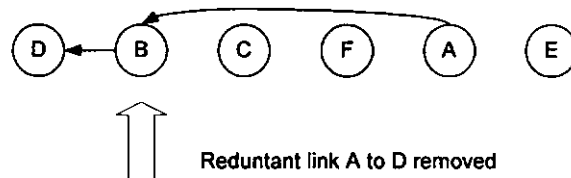


*Figure 4.18 Candidate for preceding task of Task B*



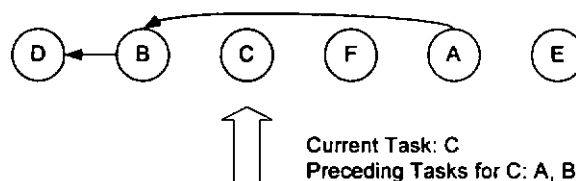
*Figure 4.19 Assignment of connection from Task A to Task B*

In this case, the link from A to D becomes a redundant link. We remove it from the graph. Note that when we select a candidate for information requirement from A to B, we do not choose D, which is a successor of A, as one of candidates. Since if we assign connection from D to B, a loop will be formed in the process structure. We get rid of such case in our algorithm by rejecting tasks that already contain information from current task as candidate for connection.

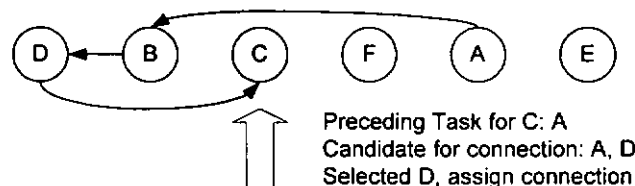


*Figure 4.20 Redundant Link Removed*

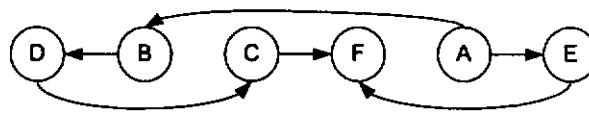
We advance the steps until all tasks in the list are processed.



*Figure 4.21 Candidates for preceding task for Task C*



*Figure 4.22 Linked assigned to Task C*



Finished result

*Figure 4.23. Finished result*

### 4.3.5. Genetic Operators Implementation

The following is a sample process specification with 20 tasks and their information dependency constraints. This example will be used intensively for the following presentation of genetic operations such as crossover, mutation and fitness evaluation.

*Table 4.1 Experiment Tasks for the algorithm*

Task No.	Name	Successors (Feed Forwards)	Successors (Feed backwards)
0	A2113 Determine markup-markdown policies	1,2,5	
1	A2121 Plan on-floor product distribution cycle and schedules	2,3,5,8,11,14,20,23	
2	A2122 Determine seasonal volume and buffers	5,6,11	
3	A2131 Develop/adapt logos and labels	4,5,6,7,8,10,11,18,19	1,
4	A2132 Design/develop visual merchandising	16,6,8,9,10,14,18,19	3,
5	A2141 Design material adoption	10,10,11,12,17,18,19,20,2,22	,3
6	A2142 Design accessories	10,10,11,12,17,18,19,20,2,22	,3
7	A2143 Design construction methods and workmanship level	10,11,14,18	4,5,6
8	A2144 Design colourway/line combos	11,12,13,14,19,22	4,5
9	A2145 Select care instructions and tag-on materials	10,12,13,14,15	,5
10	A215 Establish quality standard and policies	2,13,14,18	5,6,7
11	A216 Develop portfolio sketch books	18,19,19,2,22	1,2,5,6,7,8
12	A2311 Evaluate ad select dying processes		5,6,8,9
13	A2312 Evaluate and select add-ins finishing properties		5,6,8,9,12
14	A232 Examine overall collection image and quality consistence	18,19,2	1,5,10
15	A2331 Evaluate and select control quota categories	15,18	
16	A2332 Evaluate contemporary tariff and duty restrictions		18
17	A2333 Examine contemporary blacklisted materials & finishings		5,6
18	A2411 Fitting and sizing	20,22	7,14,15,16
19	A2412 Confirm colourways, prints, frames, silhouette	20,22	5,7,14,15
20	A2413 Streamline collection components		1,3
21	A2421 Arrange collection presentation, trade shows	22,23	0,1,2,3,11,19,20
22	A2422 Prepare catalogs	23,	3,4,5,6,8,18,19,20,2
23	A2423 Streamline collection breakdown		1,5,18,19,20

#### 4.3.5.1. Selection

Selection is the process of getting two randomly selected samples from the population to be the parents of reproduction.

*Selection*

$P_1$  := Randomly selected from population

$P_2$  := Randomly selected from population

*Figure 4.24. Selection*

#### 4.3.5.2. Crossover

The purpose of crossover in GA is to swap the genetic information of parents and hoping that a better child is produced. During crossover we randomly select tasks as crossover points. Then we swap the connection configuration of the tasks in two parents, producing two new children. Since this process changes the connection structure of the process, some information dependency constraints may no longer be fulfilled. We must apply the recover algorithm to make the children comply with the constraints. The recover algorithm will be discussed in detail in following section.

Given two randomly selected parents,  $P_1$  and  $P_2$ , from the population, the crossover operation can be formally defined as follows:

*Crossover*

$S(P_1)$  := Randomly selected positions

$S(P_2)$  := Randomly selected positions

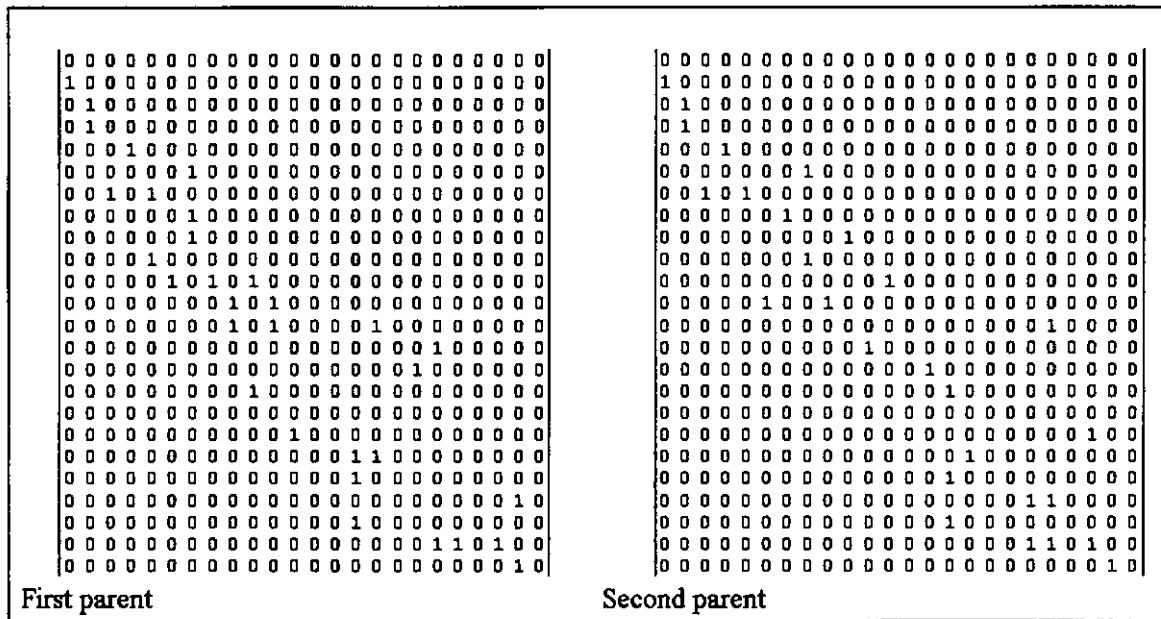
$P_1 := (P_1 - S(P_1)) \cup S(P_2)$

$P_2 := (P_2 - S(P_2)) \cup S(P_1)$

*Figure 4.25. Crossover*

*Step 1: Select two samples from population as parents for crossover*

We demonstrate the crossover by the following two parents:



*Figure 4.26. Selected parents*

*Step 2: Select points for crossover*

Points for crossover is randomly selected. To demonstrate the crossover operation in this case, we crossover the two parents at points 1, 3, 6, 9, 15, 17, 18 and 23.



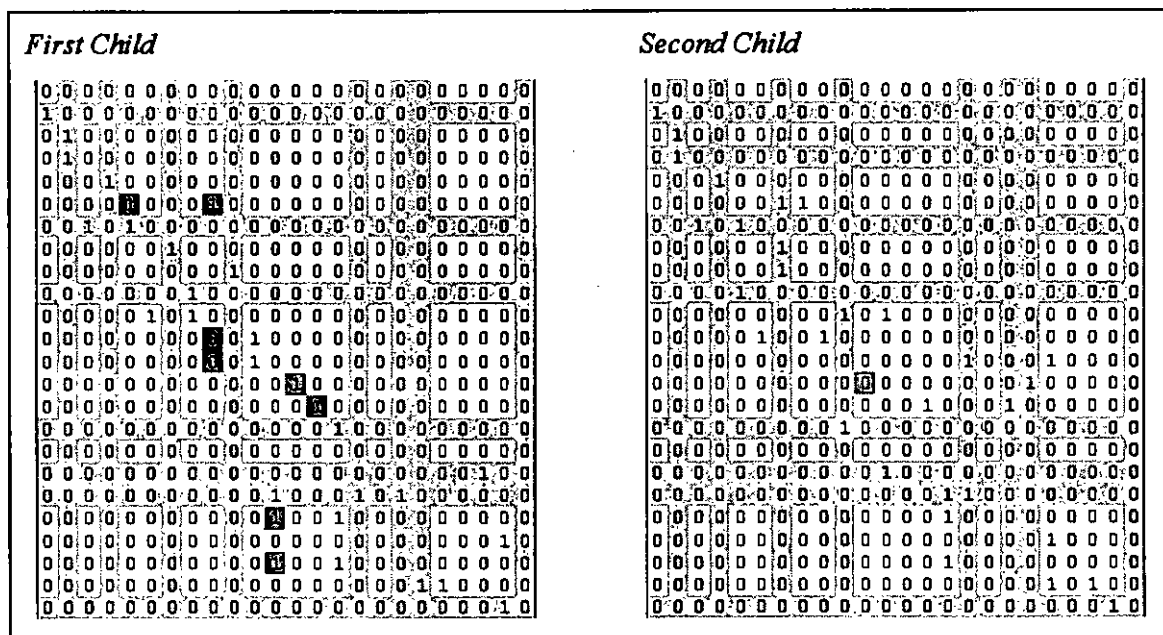


Figure 4.29. Recovery of children in reproduction

#### 4.3.5.3. Mutation

The purpose of mutation is to provide some chance for the searching points to escape from local maximum points, hence increasing the chance for finding out a global maximum fitness solution. Mutation is performed in the following steps.

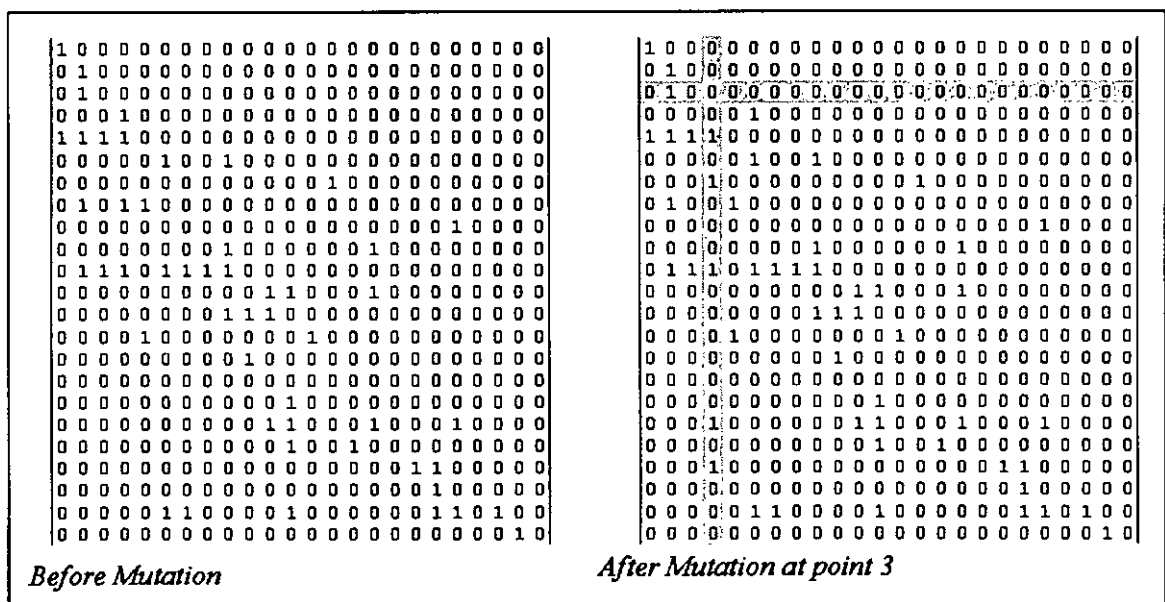


Figure 4.30. Mutation – regenerate configuration for selected tasks

### Step 1: Mutation points selection

Some tasks in the process structure are selected randomly as mutation points.

### *Step 2: Mutate the connection configurations of these points.*

For these selected tasks, clear all original connection configurations. For all preceding and succeeding tasks as specified in the dependency constraint, we randomly select from them and assign connections between the mutation point and them.

### *Step 3: Recover process structure to fulfill information dependency constraint*

Since this also breaks connections amongst between some inter-dependending tasks, we must apply recover algorithm to ensure the result is valid.

Given a randomly selected parent,  $P_1$ , from the population, the mutation operation can be formally defined as follows:

<p><i>Mutation</i></p> $M(P_1) := \text{Randomly selected positions}$ $M'(P_1) := \text{Randomly generated values for selected positions}$ $P_1 := (P_1 - M(P_1)) \cup M'(P_1)$
---

*Figure 4.31. Mutation algorithm*

## 4.3.6.Fitness and Stopping Criterion

### 4.3.6.1. Fitness Evaluation

Fitness determines how good a sample inside the population is, and usually the fitness of the sample determines the chance of survival during the evolutionary algorithm.

Since our objective of optimization is to minimize the cost incurred by feed-backwards, and the cost is usually in terms of tasks to be repeated, we must first define the cost for proceeding from one task to another in terms of resources taken for such proceeding.

If there is only one path from the destination of feed-backward to origin, the distance is simply the summation of resources taken by all tasks on this single path for proceeding from the destination to origin again, including the destination itself.

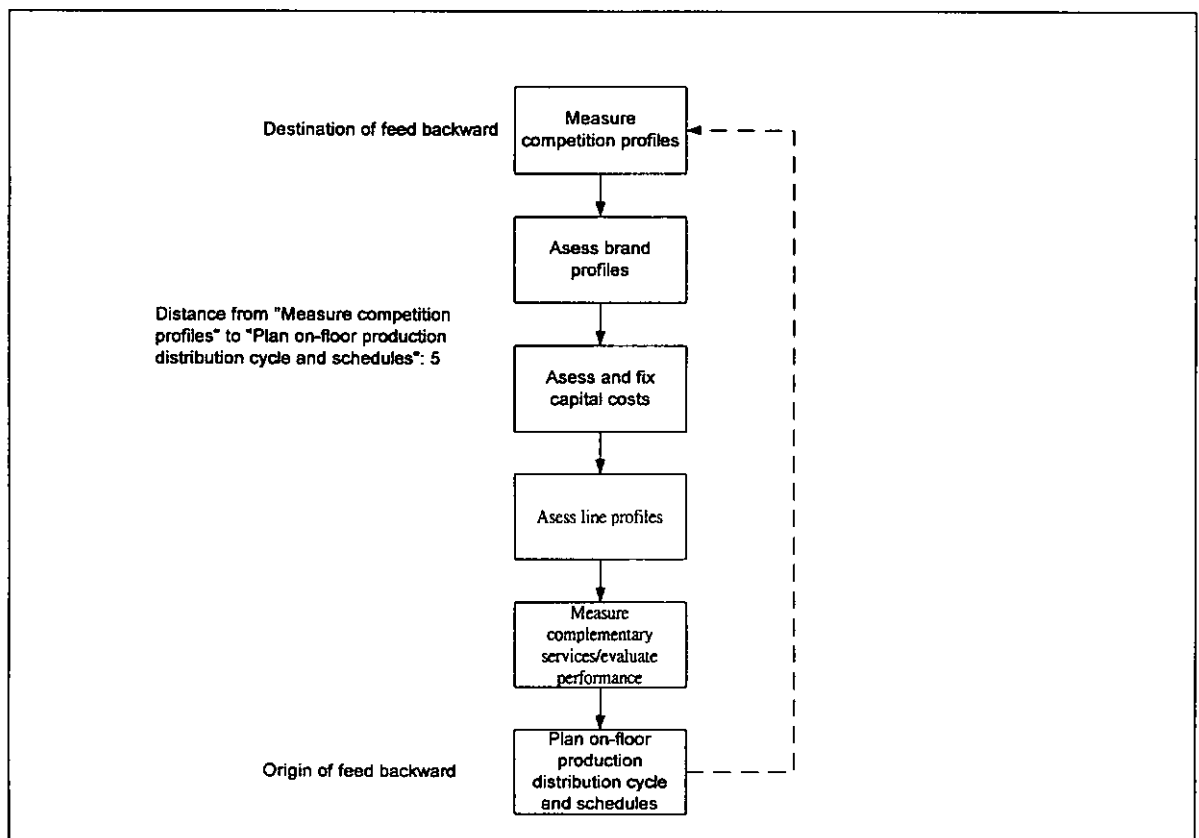


Figure 4.32 Feed back path length measurement

If there are multiple paths from the destination of feed-backward to origin, the distance is the distance of the path taking the most resources among all paths. This is because we want to measure the most significant impact of feed backwards in our evolutionary algorithm.

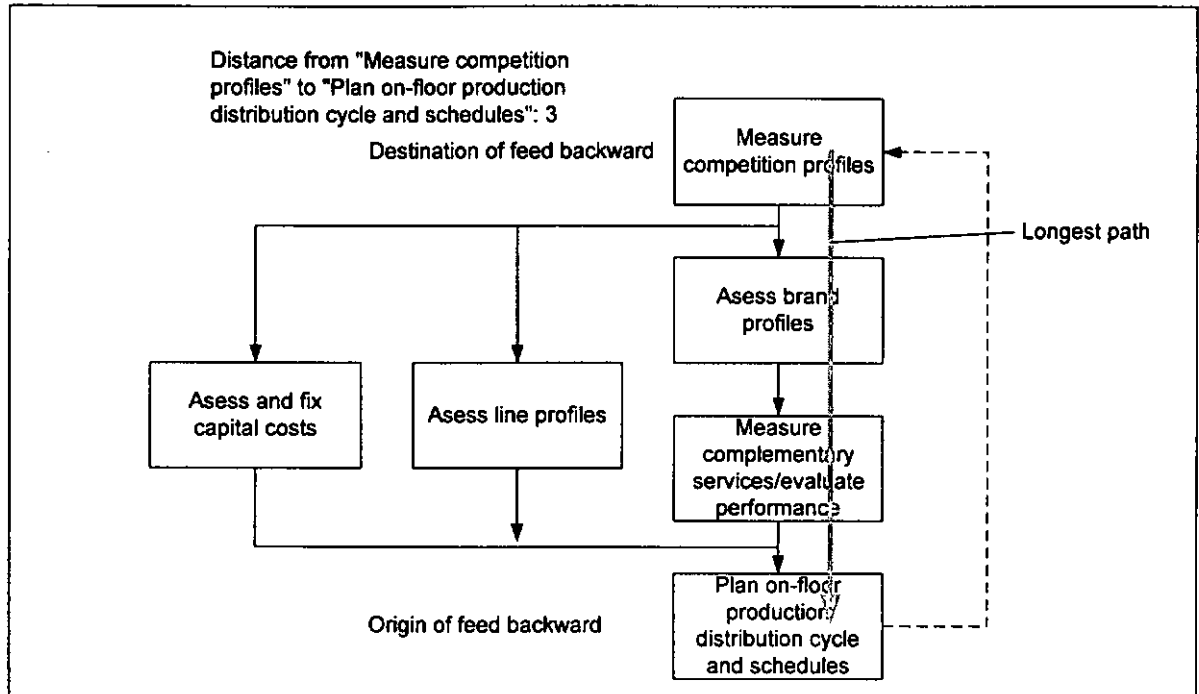


Figure 4.33 Longest Path

In the following example, the resources taken by each task are listed as follows:

Table 4.3 Resource taken for each alternative path

Task	Resources taken
Measure competition profiles	2
Assess and fix capital costs	3
Assess line profiles	2
Assess brand profiles	2
Measure complementary services/evaluate performance	2
Plan on-floor production distribution cycle and schedules	3

From this information, we can calculate the resources taken by each path:

*Table 4.3 Resource taken for each alternative path*

Path	Resources taken
Measure competition profiles -> Assess and fix capital costs -> Plan on-floor distribution cycle and schedules	5
Measure competition profiles -> Assess line profiles -> Plan on-floor distribution cycle and schedules	4
Measure competition profiles -> Assess brand profiles -> Measure complementary services/evaluate performance -> Plan on-floor distribution cycle and schedules	6

By calculation we know that the longest path is “Measure competition profiles -> Assess brand profiles -> Measure complementary services/evaluate performance -> Plan on-floor distribution cycle and schedules”.

We define the fitness of each chromosome as the total cost incurred by the longest path of all feed backward pairs of tasks.

#### **4.3.6.2. Stopping Criterion and Result**

The reproduction will stop if the number of generation is equal to a predefined stopping number. After the iteration stops, we return the best solution in the population as the solution.

# 5. Experimental Results

In this chapter, we present the results regarding the performance of the proposed algorithm for process optimization. We use Java language to implement the algorithm. A number of experiments were performed to verify the performance of the proposed algorithm and evaluate its advantage over other approaches. In section 5.2.1, we performed the experiment over simulated data, which provided fast performance for evaluation of the feasibility of the algorithm. In section 5.2.2, we performed the algorithm on real data to observe its performance under complicated cases.

## 5.1. The Experiment Details

### 5.1.1. Experimental Data

The data for experiment are the definition of tasks and precedence constraints for constructing process. Such data can be quantified by two main characteristics:

- Number of Task  $N_t$
- Number of constraints  $N_r$ , which is, the number of pair of tasks that the order for them is constrained.

The experiment runs on real and simulated data. The real data is based on a real case in the design, manufacture and distribution procedure from the textile industry. It contains 103 tasks and 926 constraints for task precedence requirement. These tasks are partitioned into 3 segments according to the approach specified in section 4.2. The simulated data is a set of 3 simplified versions of real data. They consist of the same set (and number) of tasks and with 46, 86 and 125 constraints respectively.

### **5.1.2.Objective of Experiment**

To inspect the influence of different factors to the performance of the proposed method, we compared the output of the proposed algorithm with different searching approach on different number of tasks and complexity. The different approaches used are namely random approach and greedy approach. The random approach is randomly generating the contents of the chromosomes, which is the connection matrix as in section 4.2. We will introduce the greedy approach in section 5.2.

### **5.1.3.Experiment Design**

We run the each comparison for 10 trials and take the average result. We compare the best fitness attained against the samples accessed in search space. For the greedy approach, each generation creates 1 new sample. For the genetic algorithm, approximately 8000 new solutions are created in 5000 generations. The new samples created in both search algorithms can be regarded as the items accessed in the problem space during searching. We compare the trend of the fitness versus the items accessed in the problem space during the searching process.

## **5.2.Introduction to Greedy Approach**

The greedy algorithm generates valid solutions as described in section 4.3.4. We keep the best solution generated and go on to generate another solution. If a better solution is generated, we replace the one we keep with the better one. The algorithm can be illustrated in Figure 5.1.

```

Loop
CurrentBestSolution := RandomGenerate(P)
if (Fitness(RandomGenerate(P')) > CurrentBestSolution
CurrentBestSolution := RandomGenerate(P')
Until Pre - defined run - downs

```

*Figure 5.1 The pseudo code for greedy algorithm*

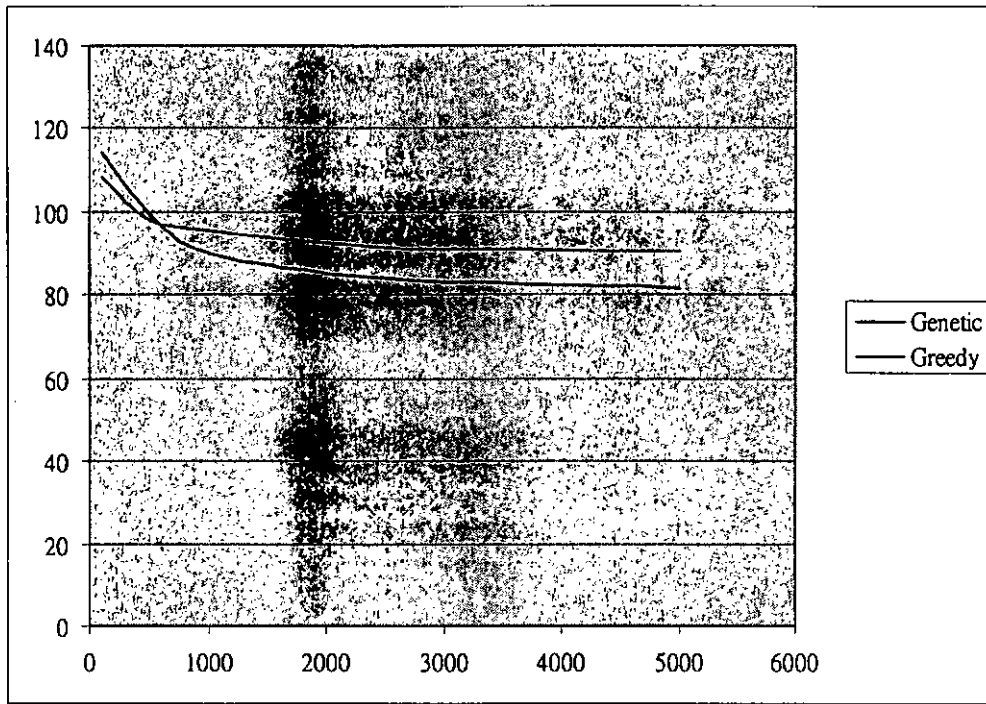
## 5.3. Performance comparison

### 5.3.1. Simulated data

Table 5.1 and Figure 5.2 shows the comparison on performance between genetic algorithm and greedy algorithm for 103 tasks and complexity of 46.

*Table 5.1 Searching performance for simulated data of 103 tasks and complexity of 46*

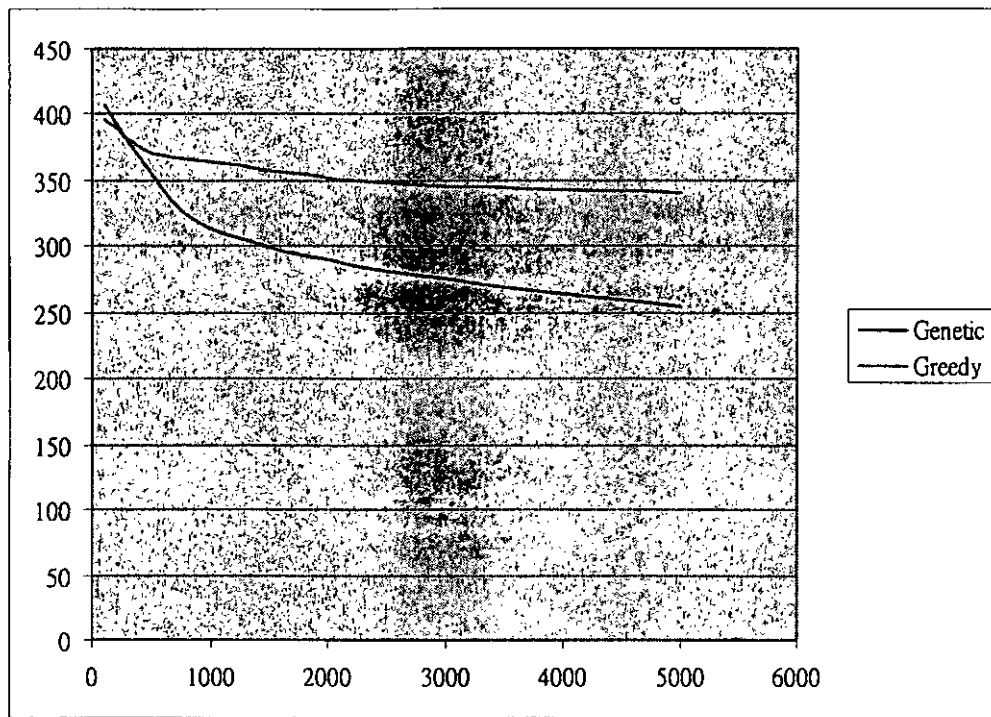
Samples accessed	Genetic	Greedy
100	114.3	108.3
500	99.3	97.8
1000	89.8	95.4
2500	84	91.7
5000	81.7	90.6



*Figure 5.2 Searching performance for simulated data of 103 tasks and complexity of 46*

*Table 5.2 Searching performance for simulated data of 103 tasks and complexity of 86*

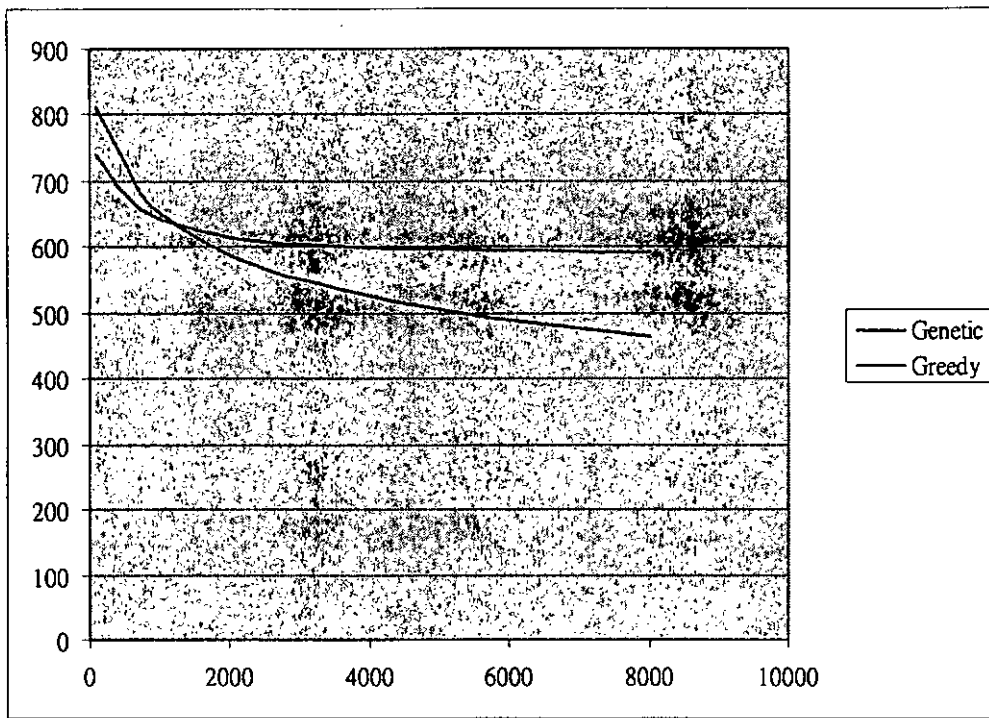
Samples accessed	Genetic	Greedy
100	406.8	395.9
500	354.9	371.2
1000	312.5	364.5
2500	281.1	347.9
5000	253.8	340.3



*Figure 5.3 Searching performance for simulated data of 103 tasks and complexity of 86*

*Table 5.3 Searching performance for simulated data of 103 tasks and complexity of 125*

Samples accessed	Genetic	Greedy
100	811.2	742.3
500	729.6	680.1
1000	652	642.4
2500	566.4	609
5000	504.1	596.4
8000	463.5	592.2



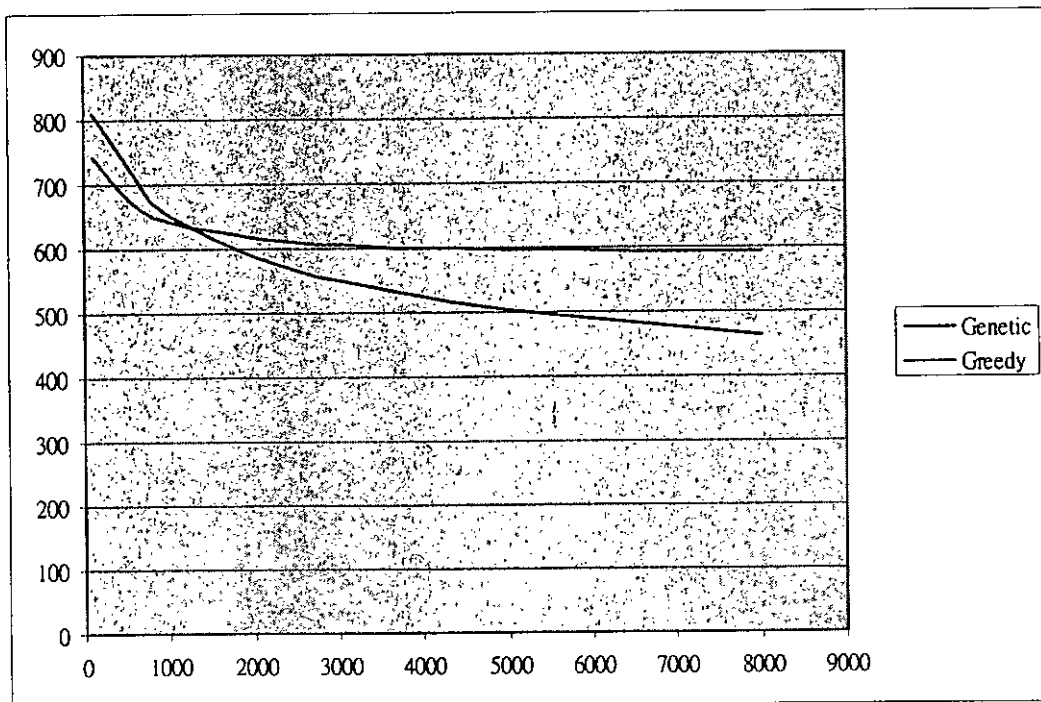
*Figure 5.4 Searching performance for simulated data of 103 tasks and complexity of 125*

### 5.3.2.Real Data

We performed a set of experiments on real workflow data from textile industry with more restrictive constraints by three methods. The comparison of the results in terms of fitness of the three methods is listed as follows.

*Table 5.4 Searching performance for real data of 36 tasks and complexity of 197*

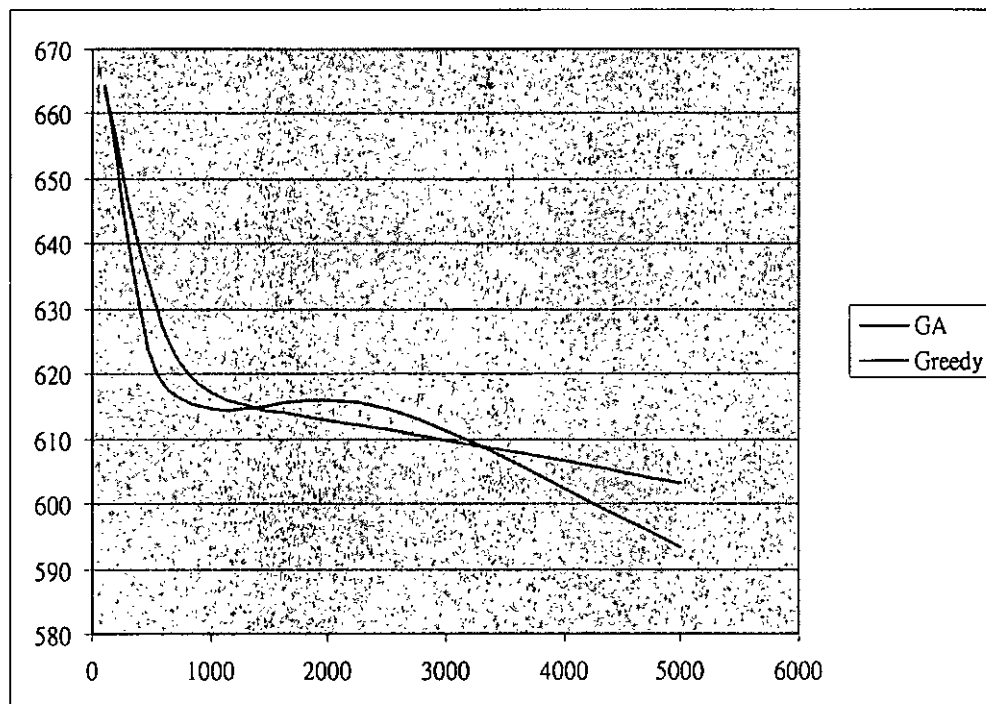
Samples accessed	Genetic	Greedy
100	811.2	742.3
500	729.6	680.1
1000	652	642.4
2500	566.4	609
5000	504.1	596.4
8000	463.5	592.2



*Figure 5.5 Searching performance for real data of 36 tasks and complexity of 197*

*Table 5.5 Searching performance for real data of 35 tasks and complexity of 245*

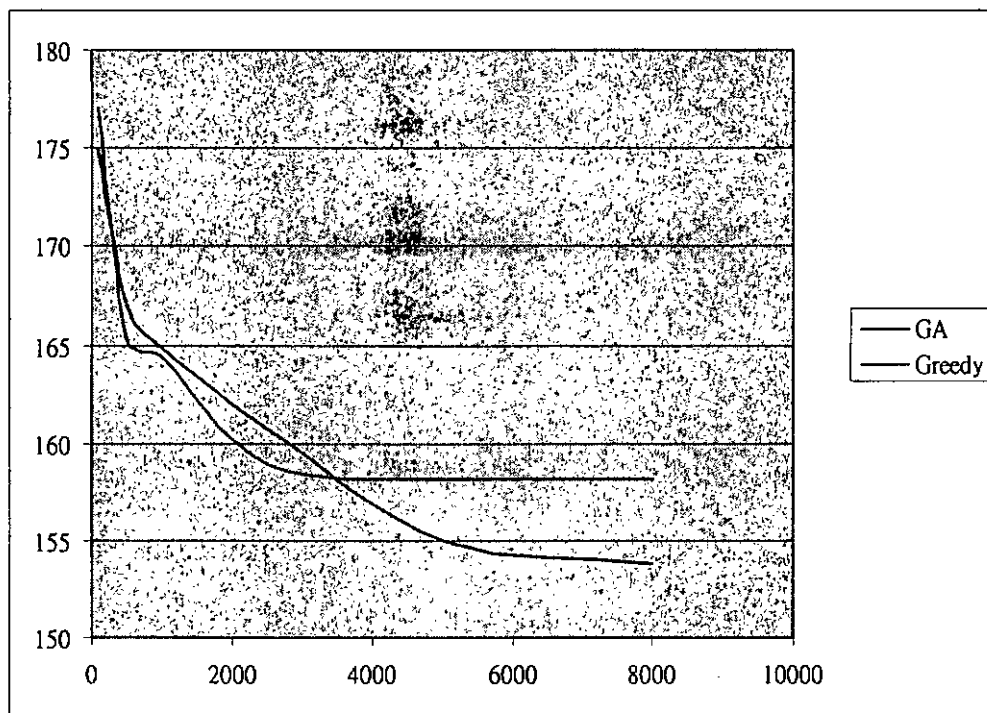
Samples accessed	Genetic	Greedy
100	664.4	663.2
500	622.2	632.6
1000	614.6	617.2
2500	614.6	611.4
5000	593.2	603.2



*Figure 5.6 Searching performance for real data of 35 tasks and complexity of 245*

*Table 5.6 Searching performance for real data of 31 tasks and complexity of 143*

Samples accessed	Genetic	Greedy
100	664.4	663.2
500	622.2	632.6
1000	614.6	617.2
2500	614.6	611.4
5000	593.2	603.2



*Figure 5.7 Searching performance for real data of 31 tasks and complexity of 143*

## **5.4. Discussion**

### **5.4.1. Advantage in performance over greedy approach**

In the experiment, we noticed that there is an advantage on the proposed approach over the greedy and pure random approaches. One difficulty in the experiment is that the absolute optimal solution of the given problem is difficult to determine. For this reason, we can only give the relative advantage of the proposed solution over other approaches. We can deduce that the main reason for such an advantage is the ability of genetic algorithm to escape from local maximal during the search iteration.

### **5.4.2. Influence on performance by process size**

At low complexity of constraints, we noticed that there is not much influence of process size on the algorithm performance. Both large and small tasks sets give similar convergence on fitness as generation grows. However, at high complexity, we discovered that the process time of the algorithm increases with the number of tasks. Moreover, given the same number of tasks, we discovered that the process time for the algorithm increases with constraint complexity.

### **5.4.3. Influence on performance by constraint**

From the experiment, we observe that the advantage for genetic algorithm over greedy approach is less for real data with more restrictive constraints. This can be explained by the fact that the search space is smaller under more restrictive constraints.

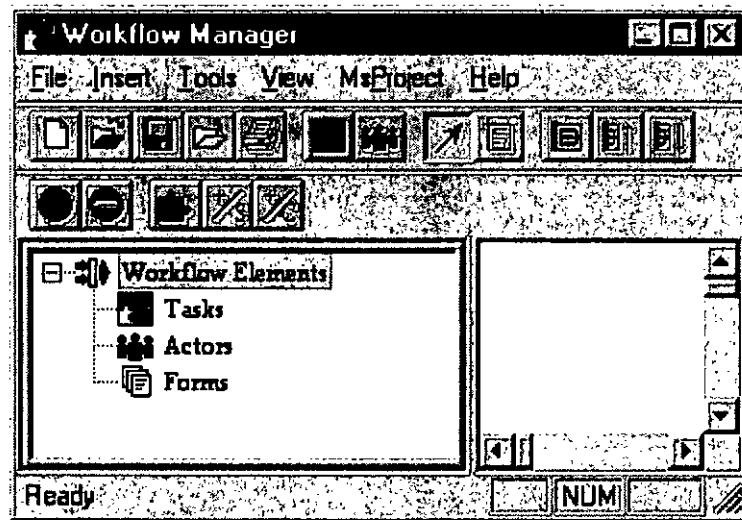
## **6. System Design And Implementation**

### **6.1. Overview of the Workflow Process Modeling tool – the Implementation Platform**

The proposed framework is implemented on process-modeling component of IPPM [Chung and Chan 99]. The process-modeling tool is implemented using Visual C++ with visual environment features [Lam, Chan, Leung and Chung 97]. It provides a graphical process editor for the user to construct a process model for automation. In [Yan and Chan 98], it employs a graphical editor to specify a workflow specification while the author employs a graphical editor for process specification. With this easy-to-use graphical user interface, processes, tasks, actors, forms and their relationships can be specified.

Process modeler can be launched to specify a process model. Under the implementation of this modeler, each process consists of tasks, actors and forms. Each task is connected by relationship elements, which are directional and indicate the flow of work. Each task is associated with actors, which are the parties responsible for performing the tasks and forms, which are the documents that the task may update or refer to.

When the process modeler is started, a graphical process editor will be displayed. With this editor, users can either specify a new process or modify an existing one.



*Figure 6.1 Workflow Modeler Interface*






The process modeler is divided into 2 panels. The process model components are classified into task, actor and form. They are displayed in an expandable tree list format on the left panel. By selecting an item on the tree list and clicking on the right mouse button, a shortcut menu will appear. There are a number of items in the shortcut menu. The properties of an item can be displayed, modified and removed by selecting and clicking on “View Property”, “Modify Property” and “Delete” menu item respectively.

On the right panel, there is a graphical process editor. In the task-based view, a user can drag and drop process components to construct a process specification. These components include start process element, end process element, task element, compulsory task relationship and optional task relationship. They can be added into the process specification by clicking on the icons in the lower toolbar and putting them on the right panel.



*Figure 6.2 Panel*

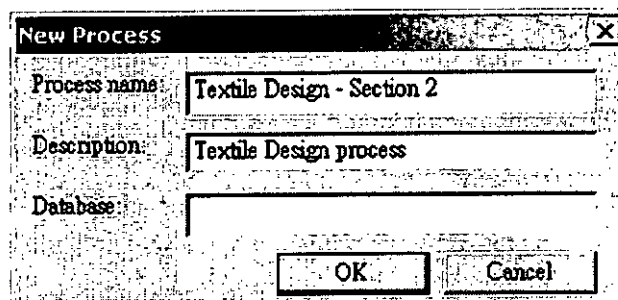
The descriptions of the icons are shown in the table below.

Icon	Description
	Insert a start process element
	Insert a end process element
	Insert a task element
	Construct a compulsory (sequential) task relationship
	Construct an optional (conditional) task relationship

*Table 6.1 Functions description*

### 6.1.1.Process

To specify a new process, the system prompts the user to give a process name, a brief description of the process and the associated application database in Lotus Notes. Actors and forms defined in Lotus Notes are imported into the process modeler for process specification. The actor information specification files and form information specification files can be selected from import actor dialogue and import form dialogue respectively. With the specification files imported, the definition of actors and forms will appear in the process modeler.



*Figure 6.3 Task properties dialog box*

When a user wants to modify an existing process definition instead of creating a new one, he/she can open a file that contains the model of a process. After the opening of an existing process definition, the system will show a graphical process specification.

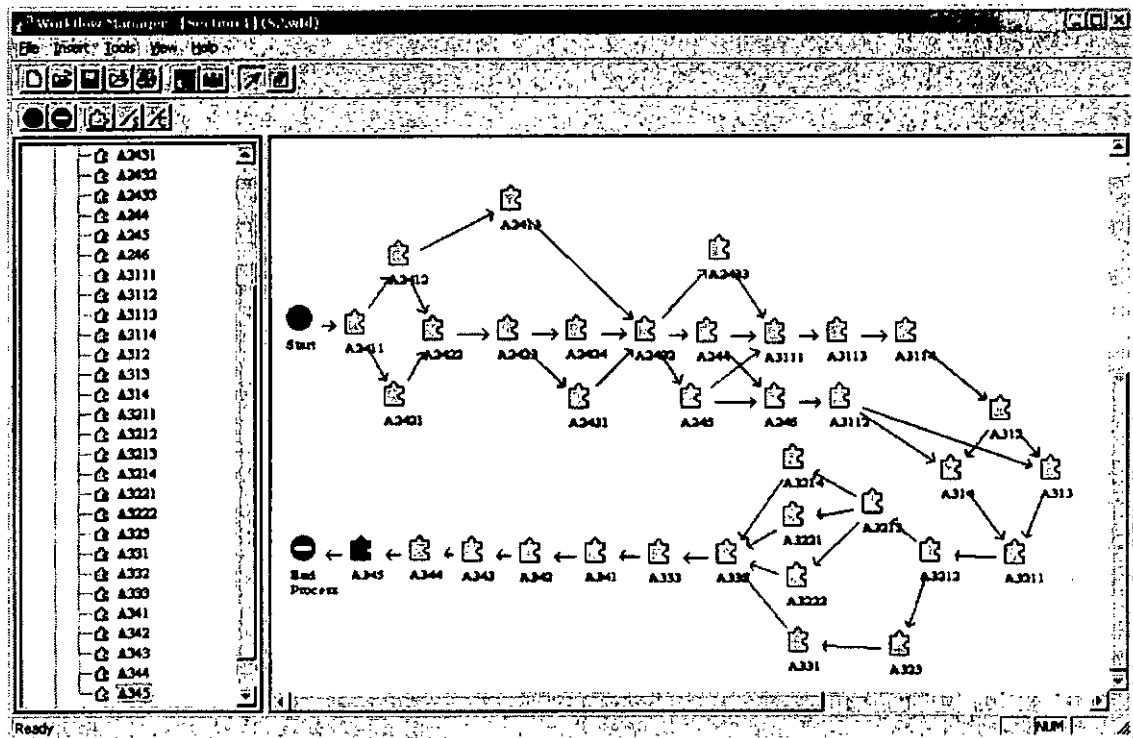
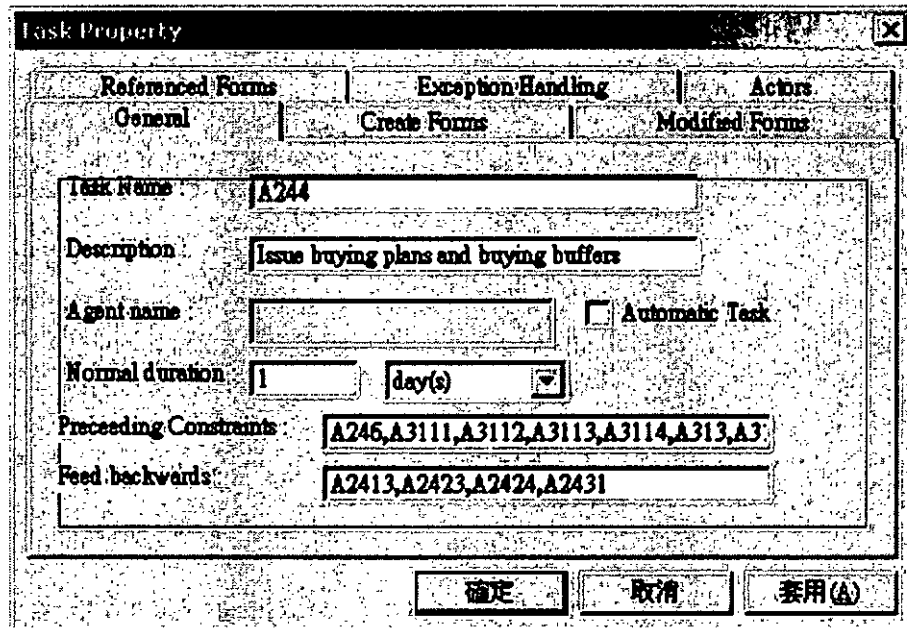


Figure 6.4 Process structure

### 6.1.2.Task

Whenever a new task is created or an existing task modified, a task property dialogue is brought up.



The image shows a 'Task Property' dialog box with a title bar and a close button. It contains three tabs: 'Referenced Forms', 'Exception Handling', and 'Actor'. The 'General' tab is selected, showing fields for 'Task Name' (A244), 'Description' (Issue buying plans and buying buffers), 'Agent name' (empty), 'Normal duration' (1 day(s)), 'Preceding Constraints' (A246,A3111,A3112,A3113,A3114,A313,A3), and 'Feed backwards' (A2413,A2423,A2424,A2431). There is an 'Automatic Task' checkbox which is unchecked. At the bottom are three buttons: '确定' (OK), '取消' (Cancel), and '套用(A)' (Apply).

Referenced Forms	Exception Handling	Actor
General	Create Forms	Modified Forms

Task Name: A244

Description: Issue buying plans and buying buffers

Agent name:

Normal duration: 1 day(s)

Preceding Constraints: A246,A3111,A3112,A3113,A3114,A313,A3

Feed backwards: A2413,A2423,A2424,A2431

☐ Automatic Task

确定 取消 套用(A)

Figure 6.5 Task properties for A244

In the task property dialog box, the user can define the task attributes, which include the name of the task and a brief description, the preceding constraints and feed-backwards information; the form it creates, modifies and references; the actor responsible for its performance and the exception handling method for it.

The flow of work among the tasks can be specified by the relationship element.

### **6.1.3.Limitation on original implementation**

The original process modeler supports manual manipulation of process structures, tasks and their associated forms and actors. Our main contribution is to provide an automatic configuration of task arrangement that gives optimal efficiency under user-defined constraints.

### **6.1.4.Concluding remarks**

There are many other sophisticated features in this process modeler,e.g. specification of mode of task, which is either manual or automatic. Forms and actors are resources associated with each task. The process modeler also supports specification and allocation of Forms and Actors for each workflow process. Since our work focuses on the process structure and concerns little the associated information for process, we will not discuss such features.

## 6.2. Adaptive workflow feature on process modeling tool

### 6.2.1. The workflow data

To demonstrate the adaptive workflow feature, we will use a workflow-tasks-and-constraints set as the demonstration data throughout this section. Here is the description of the data.

The following is a set of data for a workflow process. It lists all the tasks that should be performed under the process. It also describes the precedence constraints and feed-backward information of the tasks. In the table, the first column is the task number for identification. The second column is the name of the task. The third column indicates the precedence constraints. It contains a set of task numbers. For tasks inside the column it means that the task in this row would provide information to them. For this reason, those tasks must run after this task inside the process structure, no matter immediately following this task or following its successors.

No	Task Name	Task that should run after it	Task that it may Feed-backwards to
1	A2411 Fitting and sizing	3,5,7,10,28,30	0
2	A2412 Confirm colourways, prints, frames, silhouette	3,5,7,10,11,28,30,31	0
3	A2413 Streamline collection components	9,9,,11,12,13,31	0
4	A2421 Arrange collection presentation, trade shows	5,6,,7,8,9,10,11,13,14,17,18,19,20,21,23,24,29,30	0,2,3
5	A2422 Prepare catalogs	6,8,9,10,11	0,1,2,3,4
6	A2423 Streamline collection breakdown	7,8,9,10,11,12,13,14,16,17,18,23,24,25,27,28,30,31,32,33	0,1,2,3
7	A2424 Re-schedule material consumption	9,10,13,16,17,21,27,28,29,30,31,32	0
8	A2431 Open PDM files and item digital IDs	9,21,22,23,24,28,27,30,31,32,33	3,5,6
9	A2432 Consent delivery schedule	10,11,12,13,14,15,16,17,20,23,25,26,28,29,31,32	0,3,6
10	A2433 Conclude material specification and variation allowance	14,16,17,19,22	0,3,6
11	A244 Issue buying plans and buying buffers	13,14,15,16,17,19,20,21,22,23,24,26,28,31	0,3,6,7,8
12	A245 Determine procurement tactic and policies	13,14,16,17,18,19,20,21,22,23,24,27,28,29	0,3,6,9,10
13	A246 Determine contingent orders for market uncertainties	15,19,20,21,23,25,27,31,32,35	0,11

14	A3111 Evaluate materials/quantity availability in countries	16,17,18,20,24,29	0,9,10
15	A3112 Project uncertainty	20,25,26	0,11,13
16	A3113 Design materials/production workflow process amongst countries	17,18,20,21,23,25,26,27,29,31,35	0,10,14
17	A3114 Evaluate and optimize cost and leadtime	18,19,27,28,31,34	6,7,9,11,12,14,16
18	A312 Allocate proportion of purchase orders to potential suppliers	19,20,21,22,23,26,28,31	0,9,10,11,14,17
19	A313 Assess individual supplier performance	21,22,23,24,25,28	0,7,11,12,14,17,18
20	A314 Decide critical order placement criteria	21,22,23,24,25,27	0,3,17,18,19
21	A3211 Accredited suppliers and open supply account	22,23,26,27,28,31,31,34	3,6,8,11,19
22	A3212 Determine contractual relationships	23,24,28,29	0,11,14,15,18,19,21
23	A3213 Develop affiliated sourcing agents and offices	24,25,26,32,34,35	0
24	A3214 Examine impacts of make-or-buy decisions	29,30	3,9,14,15,21,22
25	A3221 Install communication infrastructure	29,33	0,8,10,19
26	A3222 Design communication procedures/documentation	29,33	0,8,10,19
27	A323 Assign buying teams duties and supply site visits	28,29,30,31	6,12,13,14,15,18,21,24,25
28	A331 Negotiate delivery terms - conditions	29,30,31,32,33,34,35	
29	A332 Adjust allocation of purchase orders	30,31,32,33,34,35	3,6,9,10
30	A333 Adjust order details	31,32,33,34,35	0,6,7,9,10,29
31	A341 Enter into procurement contract	32,33,34,35	
32	A342 Work out credit loan facilities	33,34,35	
33	A343 Select financial supports and estimate periodic capital return	34,35	32
34	A344 Plan credit sources (undertakings)	35	32,33
35	A345 Confirm credit issuance		31,34

Table 6.2 Sample tasks data

## 6.2.2.Initial Optimal Process Generation

The Generate process feature applies the proposed algorithm to give an optimal process structure under user defined constraints and feed-backward information. The overall procedure is defining the task first. During the definition of a task the precedence constraints are gathered from the user. Finally, the system generates the process for the user. In this section, we will look into how the feature works.

### 6.2.2.1. Gathering of precedence constraints and feed-backward information

In creating a new task or modifying an existing task, a task property dialogue is brought up. Apart from the basic information of the task, the user has to specify the precedence constraints, that is, all the tasks that must be performed before it. The user also has to specify the feed-backward information, that is, those tasks that may feed backwards to the current task when further iteration is required.

The screenshot shows a 'Task Property' dialog box with three tabs: 'Referenced Forms', 'Exception Handling', and 'Actions'. The 'Referenced Forms' tab is active, showing a 'General' sub-tab. The 'Create Forms' sub-tab is also visible. The 'Modified Forms' sub-tab is not visible. The 'Task Name' field contains 'A2132'. The 'Description' field contains 'Design/develop visual merchandising'. The 'Agent name' field is empty. The 'Automatic Task' checkbox is unchecked. The 'Normal duration' field contains '1' and 'day(s)'. The 'Preceding Constraints' field contains 'A2142, A2144, A2145, A215, A232, A2411'. The 'Feed backwards' field contains 'A2131'. A red oval highlights the 'Preceding Constraints' and 'Feed backwards' fields. At the bottom, there are three buttons: '確定' (OK), '取消' (Cancel), and '套用(A)' (Apply).

Figure 6.6 Information dependency constraints

Finally, the user defines all the tasks. The user should not specify any relationship elements and should leave the tasks unconnected.

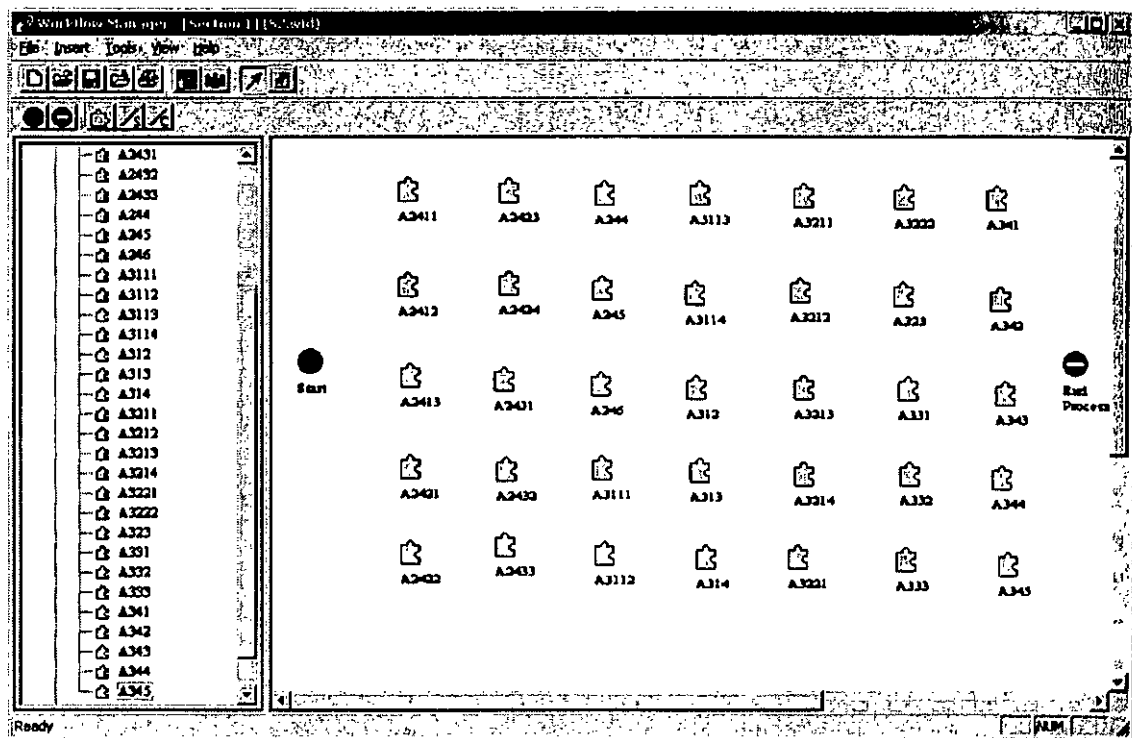
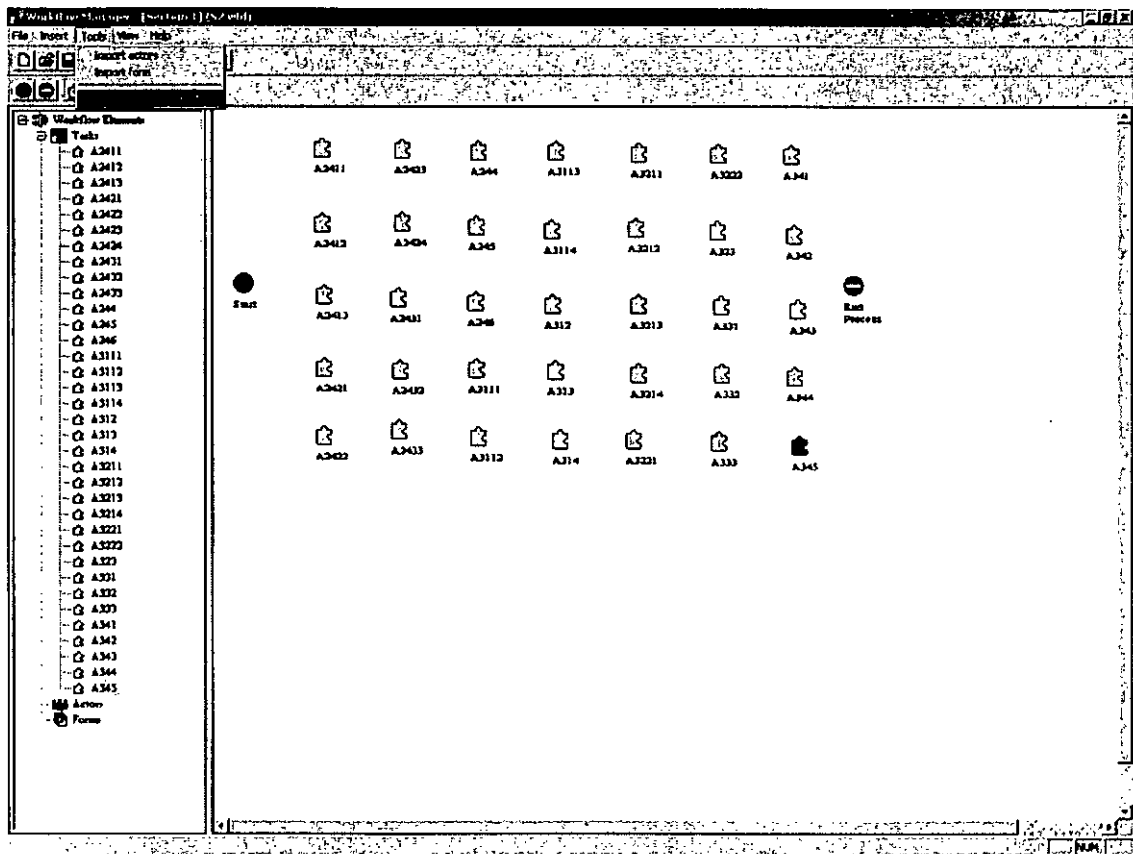


Figure 6.7 Tasks before optimization

#### 6.2.2.2. Generation the process

After all tasks in the process are defined and their precedence constraints and feed-backward information gathered, the next job is to generate the optimal process structure for the tasks. The user should click the “Generate Process Plan” command in the tools menu.



*Figure 6.8 Generating the optimal structure*

WorkFlow Manager [Scenario 1] (S2.vb6)

File Insert Tools View Help

Ready

Activity List:

- A2431
- A2432
- A2433
- A244
- A245
- A246
- A3111
- A3112
- A3113
- A3114
- A312
- A313
- A314
- A3211
- A3212
- A3213
- A3214
- A3221
- A3222
- A323
- A331
- A332
- A333
- A334
- A341
- A342
- A343
- A344
- A345

Flowchart Diagram:

The flowchart illustrates a process flow starting from a 'Start' node (black circle) and ending at a 'End Process' node (black circle with a cross). The flow is as follows:

- Start → A2411
- A2411 → A2412
- A2411 → A2421
- A2412 → A2422
- A2421 → A2422
- A2422 → A2423
- A2423 → A2424
- A2424 → A2425
- A2425 → A2426
- A2426 → A2427
- A2427 → A2428
- A2428 → A2429
- A2429 → A2430
- A2430 → A2431
- A2431 → A2432
- A2432 → A2433
- A2433 → A2434
- A2434 → A2435
- A2435 → A2436
- A2436 → A2437
- A2437 → A2438
- A2438 → A2439
- A2439 → A2440
- A2440 → A2441
- A2441 → A2442
- A2442 → A2443
- A2443 → A2444
- A2444 → A2445
- A2445 → A2446
- A2446 → A2447
- A2447 → A2448
- A2448 → A2449
- A2449 → A2450
- A2450 → A2451
- A2451 → A2452
- A2452 → A2453
- A2453 → A2454
- A2454 → A2455
- A2455 → A2456
- A2456 → A2457
- A2457 → A2458
- A2458 → A2459
- A2459 → A2460
- A2460 → A2461
- A2461 → A2462
- A2462 → A2463
- A2463 → A2464
- A2464 → A2465
- A2465 → A2466
- A2466 → A2467
- A2467 → A2468
- A2468 → A2469
- A2469 → A2470
- A2470 → A2471
- A2471 → A2472
- A2472 → A2473
- A2473 → A2474
- A2474 → A2475
- A2475 → A2476
- A2476 → A2477
- A2477 → A2478
- A2478 → A2479
- A2479 → A2480
- A2480 → A2481
- A2481 → A2482
- A2482 → A2483
- A2483 → A2484
- A2484 → A2485
- A2485 → A2486
- A2486 → A2487
- A2487 → A2488
- A2488 → A2489
- A2489 → A2490
- A2490 → A2491
- A2491 → A2492
- A2492 → A2493
- A2493 → A2494
- A2494 → A2495
- A2495 → A2496
- A2496 → A2497
- A2497 → A2498
- A2498 → A2499
- A2499 → A2500
- A2500 → A2501
- A2501 → A2502
- A2502 → A2503
- A2503 → A2504
- A2504 → A2505
- A2505 → A2506
- A2506 → A2507
- A2507 → A2508
- A2508 → A2509
- A2509 → A2510
- A2510 → A2511
- A2511 → A2512
- A2512 → A2513
- A2513 → A2514
- A2514 → A2515
- A2515 → A2516
- A2516 → A2517
- A2517 → A2518
- A2518 → A2519
- A2519 → A2520
- A2520 → A2521
- A2521 → A2522
- A2522 → A2523
- A2523 → A2524
- A2524 → A2525
- A2525 → A2526
- A2526 → A2527
- A2527 → A2528
- A2528 → A2529
- A2529 → A2530
- A2530 → A2531
- A2531 → A2532
- A2532 → A2533
- A2533 → A2534
- A2534 → A2535
- A2535 → A2536
- A2536 → A2537
- A2537 → A2538
- A2538 → A2539
- A2539 → A2540
- A2540 → A2541
- A2541 → A2542
- A2542 → A2543
- A2543 → A2544
- A2544 → A2545
- A2545 → A2546
- A2546 → A2547
- A2547 → A2548
- A2548 → A2549
- A2549 → A2550
- A2550 → A2551
- A2551 → A2552
- A2552 → A2553
- A2553 → A2554
- A2554 → A2555
- A2555 → A2556
- A2556 → A2557
- A2557 → A2558
- A2558 → A2559
- A2559 → A2560
- A2560 → A2561
- A2561 → A2562
- A2562 → A2563
- A2563 → A2564
- A2564 → A2565
- A2565 → A2566
- A2566 → A2567
- A2567 → A2568
- A2568 → A2569
- A2569 → A2570
- A2570 → A2571
- A2571 → A2572
- A2572 → A2573
- A2573 → A2574
- A2574 → A2575
- A2575 → A2576
- A2576 → A2577
- A2577 → A2578
- A2578 → A2579
- A2579 → A2580
- A2580 → A2581
- A2581 → A2582
- A2582 → A2583
- A2583 → A2584
- A2584 → A2585
- A2585 → A2586
- A2586 → A2587
- A2587 → A2588
- A2588 → A2589
- A2589 → A2590
- A2590 → A2591
- A2591 → A2592
- A2592 → A2593
- A2593 → A2594
- A2594 → A2595
- A2595 → A2596
- A2596 → A2597
- A2597 → A2598
- A2598 → A2599
- A2599 → A2600
- A2600 → A2601
- A2601 → A2602
- A2602 → A2603
- A2603 → A2604
- A2604 → A2605
- A2605 → A2606
- A2606 → A2607
- A2607 → A2608
- A2608 → A2609
- A2609 → A2610
- A2610 → A2611
- A2611 → A2612
- A2612 → A2613
- A2613 → A2614
- A2614 → A2615
- A2615 → A2616
- A2616 → A2617
- A2617 → A2618
- A2618 → A2619
- A2619 → A2620
- A2620 → A2621
- A2621 → A2622
- A2622 → A2623
- A2623 → A2624
- A2624 → A2625
- A2625 → A2626
- A2626 → A2627
- A2627 → A2628
- A2628 → A2629
- A2629 → A2630
- A2630 → A2631
- A2631 → A2632
- A2632 → A2633
- A2633 → A2634
- A2634 → A2635
- A2635 → A2636
- A2636 → A2637
- A2637 → A2638
- A2638 → A2639
- A2639 → A2640
- A2640 → A2641
- A2641 → A2642
- A2642 → A2643
- A2643 → A2644
- A2644 → A2645
- A2645 → A2646
- A2646 → A2647
- A2647 → A2648
- A2648 → A2649
- A2649 → A2650
- A2650 → A2651
- A2651 → A2652
- A2652 → A2653
- A2653 → A2654
- A2654 → A2655
- A2655 → A2656
- A2656 → A2657
- A2657 → A2658
- A2658 → A2659
- A2659 → A2660
- A2660 → A2661
- A2661 → A2662
- A2662 → A2663
- A2663 → A2664
- A2664 → A2665
- A2665 → A2666
- A2666 → A2667
- A266

93

### **6.2.3.Adaptation to changes in constraints for existing process structures**

After a process plan has been defined or even when production has already started, users may require changing the precedence constraints and feed-backward information due to the changes in business context and environment. For this reason, the existing process plan may no longer be optimal since there may exist another process structure that is superior to the current one under the new constraints. The process modeler supports re-optimizing the process structure to make it still optimal under the new constraints and to make it adapt to the new environment.

In this section, we will look into this feature for how it works.

#### **6.2.3.1. Scenario**

We will demonstrate the change in constraints by the following scenario. Recall the workflow constraint data shown in section 6.2.1. Two of the tasks, namely A244 and A245, depend on information from some other tasks. For this reason, they must go after those tasks. Now these two tasks no longer require information from those tasks. Therefore those preceding tasks should be removed from the precedence constraints. The optimal task arrangement may be changed after the removal of such constraints.

The following listing repeats the task data listed in section X.1. In the original context, task A244 provides information to tasks A246, A3111, A3112, A3113, A3114, A313, A314, A3211 A3212, A3213, A3214, A3222, A331 and A341 respectively. In the mean time, A245 provides information to A246, A3111, A3112, A3113, A3114, A313, A314, A3211 A3212, A3213, A3214, A323, A331 and A332 respectively. For this reason, all

tasks to which these two tasks provide information must go after them.

However, due to change in company policy, these two tasks no longer need to provide information to A3111, A3112, A3113, A3114 and A313. Hence, these tasks do not need to go after the two tasks anymore. Such change is marked with red lines in the following table:

No	Task Name	Task that should run after it	Task that it may Feed-backwards to
1	A2411 Fitting and sizing	3,5,7,10,28,30	0
2	A2412 Confirm colourways, prints, frames, silhouette	3,5,7,10,11,28,30,31	0
3	A2413 Streamline collection components	9,9,,11,12,13,31	0
4	A2421 Arrange collection presentation, trade shows	5,6,,7,8,9,10,11,13,14,17,18,19,20,21,23,24,29,30	0,2,3
5	A2422 Prepare catalogs	6,8,9,10,11	0,1,2,3,4
6	A2423 Streamline collection breakdown	7,8,9,10,11,12,13,14,16,17,18,23,24,25,27,28,30,31,32,33	0,1,2,3
7	A2424 Re-schedule material consumption	9,10,13,16,17,21,27,28,29,30,31,32	0
8	A2431 Open PDM files and item digital IDs	9,21,22,23,24,28,27,30,31,32,33	3,5,6
9	A2432 Consent delivery schedule	10,11,12,13,14,15,16,17,20,23,25,26,28,29,31,32	0,3,6
10	A2433 Conclude material specification and variation allowance	14,16,17,19,22	0,3,6
11	A244 Issue buying plans and buying buffers	13,14,15,16,17,19,20,21,22,23,24,26,28,31	0,3,6,7,8
12	A245 Determine procurement tactic and policies	13,14,16,17,18,19,20,21,22,23,24,27,28,29	0,3,6,9,10
13	A246 Determine contingent orders for market uncertainties	15,19,20,21,23,25,27,31,32,35	0,11
14	A3111 Evaluate materials/quantity availability in countries	16,17,18,20,24,29	0,9,10
15	A3112 Project uncertainty	20,25,26	0,11,13
16	A3113 Design materials/production workflow process amongst countries	17,18,20,21,23,25,26,27,29,31,35	0,10,14
17	A3114 Evaluate and optimize cost and leadtime	18,19,27,28,31,34	6,7,9,11,12,14,16
18	A312 Allocate proportion of purchase orders to potential suppliers	19,20,21,22,23,26,28,31	0,9,10,11,14,17
19	A313 Assess individual supplier performance	21,22,23,24,25,28	0,7,11,12,14,17,18
20	A314 Decide critical order placement criteria	21,22,23,24,25,27	0,3,17,18,19
21	A3211 Accredited suppliers and open supply account	22,23,26,27,28,31,31,34	3,6,8,11,19
22	A3212 Determine contractual relationships	23,24,28,29	0,11,14,15,18,19,21
23	A3213 Develop affiliated sourcing agents and offices	24,25,26,32,34,35	0
24	A3214 Examine impacts of make-or-buy decisions	29,30	3,9,14,15,21,22
25	A3221 Install communication infrastructure	29,33	0,8,10,19
26	A3222 Design communication procedures/documentation	29,33	0,8,10,19
27	A323 Assign buying teams duties and supply site visits	28,29,30,31	6,12,13,14,15,18,21,24,25

28	A331 Negotiate delivery terms - conditions	29,30,31,32,33,34,35	
29	A332 Adjust allocation of purchase orders	30,31,32,33,34,35	3,6,9,10
30	A333 Adjust order details	31,32,33,34,35	0,6,7,9,10,29
31	A341 Enter into procurement contract	32,33,34,35	
32	A342 Work out credit loan facilities	33,34,35	
33	A343 Select financial supports and estimate periodic capital return	34,35	32
34	A344 Plan credit sources (undertakings)	35	32,33
35	A345 Confirm credit issuance		31,34

*Table 6.3 Change in information dependence as there are changes in business environment*

Under this change, there may be another process structure that is more efficient than the existing one generated based on old constraints. We would like to find out this new optimal process structure, and replace the existing one with it.

To perform the above job, first of all, we should modify the precedence constraints for tasks A244 and A245 in the process modeler and save the changes. After this, we should apply the evolutionary algorithm again to generate a new optimal process structure. Finally, we replace the existing process structure with the newly generated one. In the coming sections, we would demonstrate how to perform this procedure.

### 6.2.3.2. Changing precedence constraints and feed-backward information on existing process structure

The user should open the process structure using the process modeler. The screen will display the existing process structure.

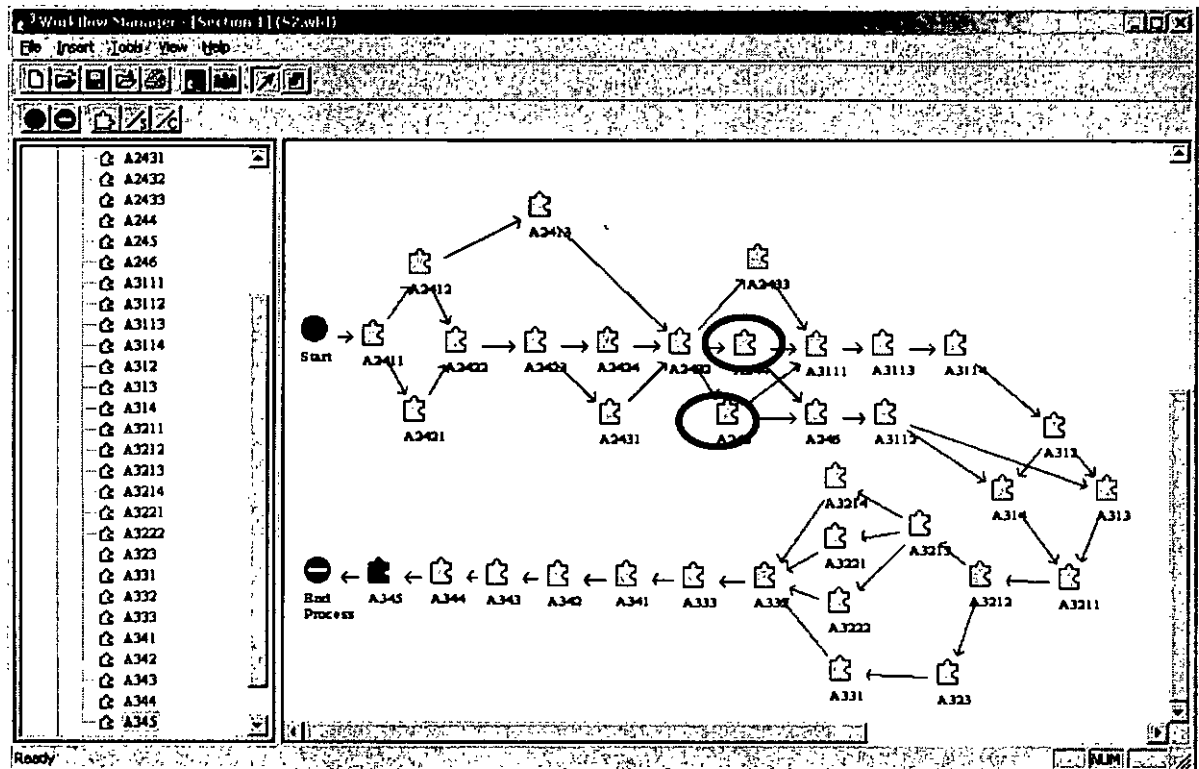


Figure 6.10 Tasks affected in change

The user should right-click the task the constraints of which he/she wants to modify. Select the “Modify properties” command in pop-up menu.



The task property dialog box should appear. The existing constraints and feed-backward information should appear. The user may edit the values and change to new constraints, as he/she desires. In this case, we edit the constraints for task A244.

Task Property	
Referenced Forms	Exception Handling
General	Create Forms
Task Name: <input type="text" value="A244"/>	
Description: <input type="text" value="Issue buying plans and buying buffers"/>	
Agent name: <input type="text"/>	<input type="checkbox"/> Automatic Task
Normal duration: <input type="text" value="1"/> <input type="text" value="day(s)"/>	
Preceding Constraints: <input type="text" value="A246,A3111,A3112,A3113,A3114,A313,A3"/>	
Feed backwards: <input type="text" value="A2413,A2423,A2424,A2431"/>	
<input type="button" value="確定"/> <input type="button" value="取消"/> <input type="button" value="套用(A)"/>	

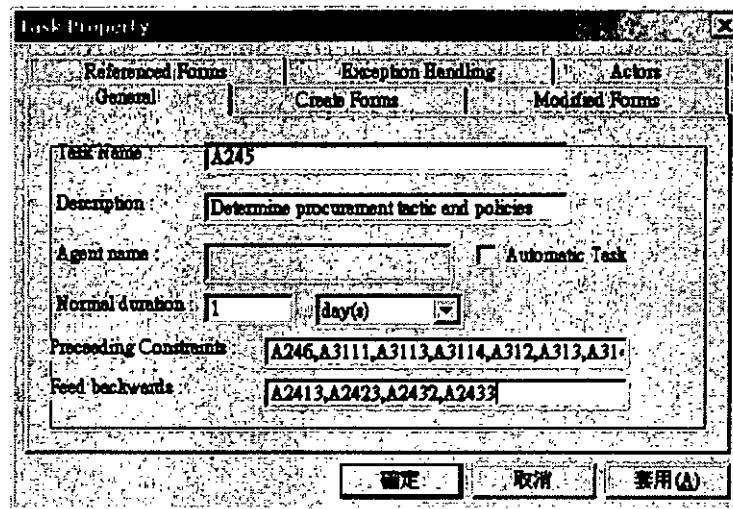
Figure 6.12 Modification for properties of task A244

We delete tasks A3111, A3112, A3113, A3114 and A313 from the precedence constraint field.

Task Property	
Referenced Forms	Exception Handling
General	Create Forms
Task Name: <input type="text" value="A244"/>	
Description: <input type="text" value="Issue buying plans and buying buffers"/>	
Agent name: <input type="text"/>	<input type="checkbox"/> Automatic Task
Normal duration: <input type="text" value="1"/> <input type="text" value="day(s)"/>	
Preceding Constraints: <input type="text" value="A246,A314,A3211,A3212,A3213,A3214,A3"/>	
Feed backwards: <input type="text" value="A2413,A2423,A2424,A2431"/>	
<input type="button" value="確定"/> <input type="button" value="取消"/> <input type="button" value="套用(A)"/>	

Figure 6.13 Modification for properties of task A244

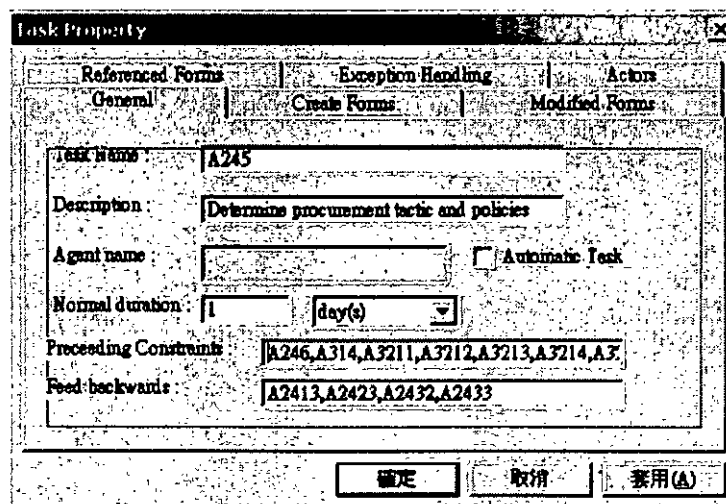
Then we also double-click on task A245 in the process plan screen. The task properties are displayed.



Task Property	
Referenced Forms	Exception Handling
General	Create Forms
<p>Task Name: A245</p> <p>Description: Determine procurement tactic and policies</p> <p>Agent name: <input type="text"/> <input type="checkbox"/> Automatic Task</p> <p>Normal duration: 1 day(s)</p> <p>Preceding Constraints: A246,A3111,A3113,A3114,A312,A313,A31</p> <p>Feedbacks: A2413,A2423,A2432,A2433</p>	
<p>確定 取消 套用(A)</p>	

Figure 6.14 Modification for properties of task A245

Then we delete tasks A3111, A3113, A3114, A312, A313 from the precedence constraints field.



Task Property	
Referenced Forms	Exception Handling
General	Create Forms
<p>Task Name: A245</p> <p>Description: Determine procurement tactic and policies</p> <p>Agent name: <input type="text"/> <input type="checkbox"/> Automatic Task</p> <p>Normal duration: 1 day(s)</p> <p>Preceding Constraints: A246,A314,A3211,A3212,A3213,A3214,A3</p> <p>Feedbacks: A2413,A2423,A2432,A2433</p>	
<p>確定 取消 套用(A)</p>	

Figure 6.15 Modification for properties of task A245

After finishing editing, the user should click close button to close and save the changes.

Or if the user wants to perform other commands after editing, he/she can click apply button to save the changes without closing the dialog box.

### 6.2.3.3. Re-optimize the workflow

The re-optimization of the workflow process structure makes it adapt to the changes in constraints. After the constraints for the tasks are modified and the generate-process-structure command is re-issued, a new process structure, optimal under the new constraints, will be generated to replace the existing one.

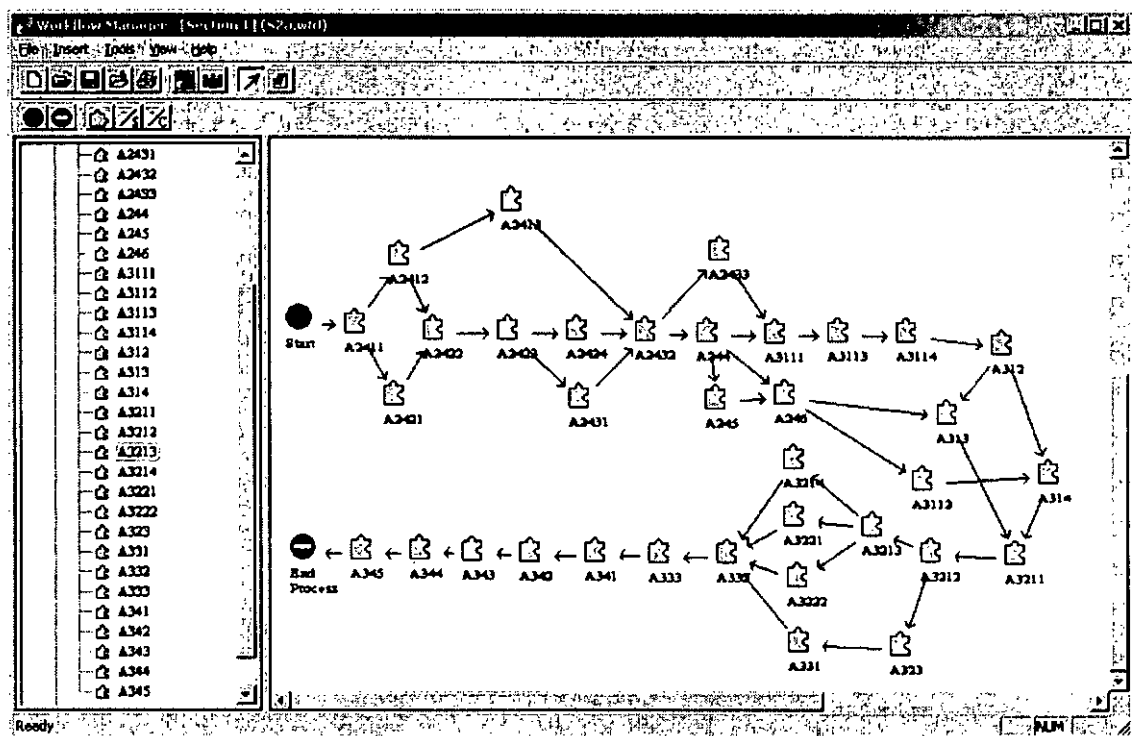


Figure 6.16 Re-generation of optimal process structure after changes in constraints

# 7. Conclusion

## 7.1. Summary

Modeling and managing business process with workflow technology had be developed and implemented for long time and the technology had become matured. Such workflow consists of process, actors and forms. Processes in workflow are consists of tasks. Such tasks are connected by the precedence relationships. Under predefined criteria such tasks may be completed and invoked other successors. We had proposed the geno-type representation of such tasks relationships. For this reason, the optimization of process structure with evolutionary algorithm becomes possible. We also studied the model of business decision iterations and identified information feed-backwards as the main source of in-efficiency. This discovery provides the direction of our work to optimize workflow by minimizing the delay due to such information feed-backwards.

The evolutionary approach for optimization of inter-organizational workflow process we had presented improves process structures and task arrangement while maintaining its compliance to the user-defined business constraints. The proposed method automates the searching of the optimal task arrangement and eliminates the demand of human decision together with the excessive effort and potential human error that may come with it. The job of minimizing the delay due to feed-backwards that exists during the iteration cycles for business decisions is automated.

Its performance is superior to other optimization approaches. Although the advantage in performance depends on the complexity of constraints on process structures, we believe that this limitation is reasonable. Because more restrictive constraints means less

alternative ways for achieving the business goal, and hence the less the advantage in performance for optimization, which is the process of finding out the optimal way among many alternatives.

### **7.1.1.Unique feature of the proposed framework**

There are readings on speed up of workflow instance in [Kafeza and Karlapalem 01]. However they have no strategic planing for coordination in inter-organizational environment. Existing workflow solution cannot estimate the delay incurred by decision iterations. The proposed framework can minimize such delay.

## **7.2.Future work**

Further enhancements will be on extending our tool to organization level management. It can be applied on inter-organizational activities management for the e-commerce applications.

### **7.2.1.Application on e-commerce application**

Inter-organizational business activities often involve complicated communication channels and the way that information is passed among business parties is difficult to trace and analysis. For this reason, many communication overheads are incurred and make the work inefficient. Moreover, it is difficult to optimize the workflow process to get rid of the problem. Our proposed framework may extend to handle inter-organization business process and optimize the process for better performance.

The e-commerce changes the ways for operation of traditional organization. [Geisler 01] stated 4 dimensions for structural analysis of organization

- Formalization
- Departmentalization
- Centralization
- Complexity

In migration to e-commerce from traditional commerce, the impact to the above dimensions must be considered. Our proposed framework provides a control to inter-organizational activities. The formal specification of preceding and succeeding tasks provides the formalization. The segmentation provides the departmentalization concepts. The global re-arrangement of workflow tasks provides centralization. Our framework is capable for optimization of complicated tasks.

Inter-organizational business activities often involve complicated communication channels and the way that information is passed among business parties is difficult to trace and analysis. For this reason, many communication overheads are incurred and make the work inefficient. Moreover, it is difficult to optimize the workflow process to get rid of the problem. Our proposed framework may extend to handle inter-organization business process and optimize the process for better performance.

#### **7.2.1.1. Inter-organizational activities support**

Definition of organizational boundary and communication support may be augmented to the proposed framework. This enables support of e-enterprise on worldwide scale. It is important for e-commerce applications.

### **7.2.2.Use of agents**

Agent framework may be incorporated into the proposed framework. Intelligent agents can implement the decision logic and mobility of inter-organizational business.

[Yan, Maamar and Shen 01] states two approaches for applying agent technology to workflow management systems, namely agent-enhanced and agent-based. Agent enhanced approach means that a large centralized system with functionality implements as agents. For this reason, the agents under this approach do not know how to communicate with other heterogeneous systems. Agent based approach means to employ multi-agent system basis that binds multiple workflow management system together. This can bind the WFMS of different organizations to form an inter-organizational infrastructure for e-commerce applications.

### **7.2.3.Data Warehouse and Knowledge Management**

Our framework exploited relationship between the arrangement of tasks and performance. With such basis, data mining techniques such as association may be applied to the optimization results. The result of mining provides an important knowledge for organization to improve their business.

### **7.2.4.Application on Supply chain management**

Global competition and rapidly changing requirement is the characteristic of the environment for supply chain management. Workflow technology provides a good way for organization to management their work. However traditional workflow management systems are often criticized for they are too rigid and cannot adapt to the changing environment. Our framework can provide workflow management the adaptability for

this changing environment.

## 8. References

- [Ali, Nakao, Tan and Chen 99] Ali, F.F., Nakao, Z., Tan, R.B. and Chen, Y.W. An evolutionary approach for graph coloring, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC '99)*, Tokyo, Japan, 12-15 Oct. 1999, vol.5, pp.527-532
- [Ali, Ramamurthy and Deogun 99] Ali, M., Ramamurthy, B. and Deogun, J.S. Routing algorithms for all-optical networks with power considerations: the unicast case, *Proceedings of the Eight International Conference on Computer Communications and Networks*, Boston, MA, USA, 11-13 Oct. 1999, pp.237-241
- [Andreoli, Pacull & Pareschi 00] Andreou, J. Pacull, F. and Pareschi, R. XPECT: a framework for electronic commerce, *IEEE Internet Computing*, vol.1, issue 4, pp. 40-48, 1997
- [Barreto and Barbosa 01] Barreto, A.M.S. and Barbosa, H.J.C. Graph layout using a genetic algorithm, *Proceedings of the Sixth Brazilian Symposium on Neural Networks*, Rio de Janeiro, RJ, Brazil, 22-25 Nov. 2000, pp.179-184
- [Bider and Khomyakov 00] Bider, I. and Khomyakov, M. Is it possible to make workflow management systems flexible? Dynamical systems approach to business processes, *Proceedings of Sixth International Workshop on Groupware*, Madeira, Portugal, 18-20 Oct. 2000, pp.138-141
- [Borgida and Murata 00] Borgida, A. and Murata, T. Handling of irregularities in human centered systems: a unified framework for data and processes, *IEEE Trans. on Software Engineering*, vol. 26, Issue 10, Oct 2000, pp.959-977
- [Chang, Gautama and Dillion 01] Chang, E. Guatama, E. and Dillon, T.S. Extended Activity diagrams for Adaptive Workflow Modelling, *Proceedings of Fourth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)*, Magdeburg, Germany, 2-4 May 2001, pp.413-419
- [Chen, Fu and Lin 00] Chen, J.H., Fu, L.C. and Lin, M.H. Petri-net and GA based approach to modeling, scheduling, and performance evaluation for wafer fabrication, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, San Francisco, CA, USA, 24-28 April 2000, vol.4, pp.3403-3408
- [Choi and Moon 01] Choi, S.S. and Moon, B.R. A graph-based approach to the sorting network problem, *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, South Korea, 27-30 May 2001, pp.457-464
- [Chung and Chan 99] Chung, L.M.L. and Chan, K.C.C. "Integrating Project Planning and Process Modeling for Software Development", *IEEE Symposium on Application-Specific System and Software Engineering & Technology*, Richardson, TX, USA, 24-27 March, 1999, pp.276-279

[DeFazio, Krishnan, Srinivasan & Zeldin 01] DeFazio, S. Krishnan, R. Srinivasan, J. and Zeldin, S. The importance of extensible database systems for e-commerce, *Proceedings of 17th International Conference on Data Engineering*, Heidelberg, Germany, 2-6 April, 2001, pp.63-70

[Eckert, Claudia M; Cross, Nigel and Johnson, Jeffrey 00] Eckert, Claudia M; Cross, Nigel and Johnson, Jeffrey, Intelligent Support for Communication in Design Teams: Garment Shape specification in the Knitwear Industry, pp99-112, I, vol.21, issue 1, 2000

[Ellis, Keddra and Rozenberg 95] Ellis, C. Keddra, K. and Rozenberg, G. Dynamic Changing within Workflow Systems, pp. 10-21, *Proceedings of conference on Organizational computing systems*, 1995.

[Fox 94] Fox, M.S. ISIS: A Retrospective. In Fox, M.S. and Zweben, M., eds., *Intelligent Scheduling*, MorganKaufmann Publishers, San Francisco, California, U.S.A. pp.3-28

[Gal & Montesi 99] Gal, A. and Montesi, D. Inter-enterprise workflow management systems, *Proceedings of Tenth International Workshop on Database and Expert Systems Applications*, Florence, Italy, 1-3 Sept. 1999, pp.623-627

[Geisler 01] Geisler E., Organizing for e-business: the implementation of management principles in electronic commerce, *Proceedings of the Portland International Conference on Management of Engineering and Technology (PICMET '01)*, Portland, OR, USA, 29 July-2 Aug. 2001 vol.2, pp.22-28

[Glance, Pagani and Pareschi 96] Glance, N.S. Pagani, D.S. and Pareschi, R. Generalized process structure grammars GPSG for flexible representations of work, *Proceedings of the ACM 1996 conference on Computer supported cooperative work*, Nov. 1996, pp.180-189

[Grigori, Charoy and Godart 01] Grigori, D. Charoy, F. and Godart, C. Flexible data management and execution to support cooperative workflow: the COO approach, *The Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications, (CODAS 2001)*, Beijing, China, 23-24 April 2001, pp.124-131

[Haake and Wang 97] Haake, J.M. and Wang, W. Flexible support for business processes: extending cooperative hypermedia with process support, *Proceedings of the international ACM SIGGROUP conference on Supporting group work: the integration Challenge*, Phoenix, Arizona, United States, 1997, pp.341-350

[Halliday, Shrivastava and Wheeler 01] Halliday, J.J. Shrivastava, S.K. and Wheeler, S.M. Flexible Workflow Management in the OPENflow System, *Proceedings of Fifth IEEE International Enterprise Distributed Object Computing Conference*, Seattle, WA, USA, 4-7 Sept. 2001, pp.82-92

[Han, Sheth and Bussler 98] Han, Y. Sheth, A. and Bussler, C. A Taxonomy of Adaptive Workflow Management, <http://ccs.mit.edu/klein/cscw98/paper03/> 1998

[Hayes, Peyrovian, Sarin, Schmidt, Swenson & Weber 01] Hayes, J.G. Peyrovian, E. Sarin, S. Schmidt, M.T. Swenson, K.D. and Weber, R. Workflow interoperability standards for the Internet, *IEEE Internet Computing*, vol. 4, issue 3, pp. 37-45, 2001

- [Heinl, Horn, Jablonski, Neeb, Stein and Teschke 99] Heinl, P. Horn, S. Jablonski, S. Neeb, J. Stein, K. and Teschke, M. A comprehensive approach to flexibility in workflow management systems, *Proceedings of the international joint conference on Work activities coordination and collaboration*, San Francisco, California, USA, 22-25 Feb. 1999, pp.79-88
- [Hoffner, Lwdwig, Gulcu and Grefen 00] Hoffner, Y. Ludwig, H. Gulcu, C. and Grefen, P. An architecture for cross-organisational business processes, *Second International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000)*, Milpitas, CA, USA, 8-9 June, 2000, pp.2-11
- [Huth, Smolnik and Nastansky 01] Huth, C. Smolnik, S. and Nastansky, L. Applying topic maps to ad hoc workflows for semantic associative navigation in process networks, *Proceedings of Seventh International Workshop on Groupware*, Darmstadt, Germany, 6-8 Sept. 2001, pp.44-49
- [Ivanovic and Budimac 99] Ivanovic, M. and Budimac, Z. A framework for agent-oriented workflow in agriculture, *Proceedings. Third International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '99)*, New Delhi, India, 23-26 Sept. 1999, pp.18-22
- [Jassawalla A.R. and Sashittal, H.C. 98] Jassawalla, A.R. and Sashittal, H.C. An Examination of Collaboration in High Technology New Product Development Process, *Journal of Product Innovation Management*, Vol. 15, no. 3, pp.237-254. 1998
- [Joeris 97] Joeris, G. Cooperative and integrated workflow and document management for engineering applications, *Proceedings of Eighth International Workshop on Database and Expert Systems Applications*, Toulouse, France, 1-2 Sept. 1997, pp.68-73
- [Kafeza and Karlapalem 01] Kafeza E. and Karlapalem, K. A framework for speeding up workflow instances by exploiting alternate paths, *Proceedings of Third International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS)*, San Juan, CA, USA, 21-22 June 2001, pp.136-143
- [Knowles and Corne 00] Knowles, J. and Corne, D. A new evolutionary approach to the degree-constrained minimum spanning tree problem, *IEEE Transactions on Evolutionary Computation*, vol. 4, Issue 2, pp.125-134, 2000
- [Kuntz, Lehn and Briand 00] Kuntz, P., Lehn, R. and Briand, H. Dynamic rule graph drawing by genetic search, *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, TN, USA, 8-11 Oct. 2000, vol.4, pp.2481-2486
- [Lam, Chan, Leung and Chung 97] Lam, S.W.C. Chan, K.C.C. Leung, H.K.N. and Chung, L.M.L. A Visual Programming Environment for Z Specification, *Proceedings of the Asia-Pacific Software Engineering Conference*, Hong Kong, 1997, pp. 511-512
- [Landwehr 99] Landwehr, B. A genetic algorithm based approach for multi-objective data-flow graph optimization, *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '99)*, Wanchai, Hong Kong, 18-21 Jan. 1999, vol.1, pp.355-358

- [Le Pallec and Vantrois 01] Le Pallec, X. and Vantrois, T. A Cooperative Workflow Management System with the Meta-Object Facility, *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, Seattle, WA, USA, 4-7 Sept. 2001, pp.273-280
- [Lee, Han and Lee 01] Lee, S.I. Han, D. and Lee, D. Supporting voluntary disconnection in WFMSs, *The Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications (CODAS 2001)*, Beijing, China, 23-24 April 2001, pp.132-139
- [Lee, Han and Shim 01] Lee, M. Han, D. and Shim, J. Set-based access conflicts analysis of concurrent workflow definition, *The Proceedings of the Third International Symposium on Cooperative Database Systems for Advanced Applications (CODAS 2001)*, Beijing, China, 23-24 April 2001, pp.172-176
- [Lenz and Oberweis 01] Lenz, K. and Oberweis, A. Modeling Interorganizational Workflow with XML Nets, *Proceedings of the 34<sup>th</sup> Hawaii International Conference on System Sciences*, Hawaii, Jan. 3-6, 2001, pp.2589-2598
- [Ling and Loke 01] Ling, S. and Loke, S.W. Verification of Itineraries for Mobile Agent Enabled Interorganizational Workflow, *Proceedings of 12<sup>th</sup> International Workshop on Database and Expert Systems Applications*, Munich, Germany, 3-7 Sept. 2001, pp.582-586
- [List, Schiefer and Bruckner 01] List, B. Shiefer, J. and Bruckner, R.M. Measuring knowledge with workflow management systems, *Proceedings of 12<sup>th</sup> International Workshop on Database and Expert Systems applications*, Munich, Germany, 3-7 Sept. 2001, pp.467-471
- [Lucertini, Pacciarelli and Pacifici 99] Lucertini, M. Pacciarelli, D. and Pacifici, A. Integrating Layout Design and Material Flow Management in Assembly Systems, *Modeling Manufacturing Systems*, Berlin, New York pp.45-64, 1999
- [Magyar, Johnsson and Nevalainen 00] Magyar, G., Johnsson, M. and Nevalainen, O. An adaptive hybrid genetic algorithm for the three-matching problem, *IEEE Transactions on Evolutionary Computation*, vol4, Issue 2, pp.135-146, 2000
- [Marazakis, Papadakis and Nikolaou 98] M. Marazakis, D. Papadakis and C. Nikolaou, Management of work sessions in dynamic open environments, *Proceedings of Ninth International Workshop on Database and Expert Systems Applications*, Vienna, Austria, 26-28 Aug. 1998, pp.725-730
- [Milne 00] Milne, A.J. Analysing the Activity of Multidisciplinary Teams in the Early Stages of Conceptual Design: Method and Measure, in Scrivener, Ball and Woodcock (eds), *Collaborative Design*, Springer-Verlag London, ISBN: 1852333413. 2000
- [Papa, Bremer, Hale & Shenoit 01] Papa, M. Bremer, O. Hale, J. and Shenoit, S. Formal analysis of e-commerce protocols, *Proceedings of 5th International Symposium on Autonomous Decentralized Systems*, Dallas, TX, USA, 26-28 March 2001, pp.19-28

- [Pour 99] Pour, G. Enterprise JavaBeans, JavaBeans and XML expanding the possibilities for Web-based enterprise application development, *Proceedings of Technology of Object-Oriented Languages and Systems, TOOLS 31*, Nanjing, China, 22-25 Sept. 1999, pp. 282-291
- [Qian 99] Qian, Y. Image interpretation with fuzzy-graph based genetic algorithm, *Proceedings of the International Conference on Image Processing (ICIP 99)*, Kobe, Japan, 24-28 Oct. 1999, vol.1, pp.545-549
- [Sawionek, Wojciechowski and Arabas 99] Sawionek, B., Wojciechowski, J. and Arabas, J. Designing regular graphs with the use of evolutionary algorithms, *Proceedings of the 1999 Congress on Evolutionary Computation (CEC 99)*, Washington, DC, USA, 6-9 July 1999, vol.3, p.1839
- [Schiffmann 00] Schiffmann, W. Encoding feedforward networks for topology optimization by simulated evolution, *Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, Brighton, UK, 30 Aug.-1 Sept. 2000, pp.361-364
- [Schulz and Orlowska 01] Schulz, K. and Orlowska, M.E. Architectural issues for cross-organisational B2B interactions, *Proceedings of the International Conference on Distributed Computing Systems Workshop*, Mesa, AZ, USA, 16-19 April 2001, pp.79-87
- [Tabbara, Dana and Mansour 00] Tabbara, H., Dana, T. and Mansour, N. Heuristics for graph decomposition, *Proceedings of the 7th IEEE International Conference on Electronics, Circuits and Systems (ICECS 2000)*, Jounieh, Lebanon, 17-20 Dec. 2000, vol.2, pp.650-653
- [Tang 99] Tang, J. Ad-Hoc recovery in workflows, a formal model and some system support aspects, *Proceedings of 1999 IFCIS International Conference on Cooperative Information Systems*, Edinburgh, UK, 2-4 Sept. 1999, pp.222-233
- [Tang and Xing 99] Tang, J. and Xing, X. A workflow management systems architecture that support ad-hoc recoveries, *Proceedings of IDEAS '99, International Symposium on Database Engineering and Applications*, Montreal, Que., Canada, 2-4 Aug. 1999, pp.332-340
- [Tang, Eshraghian and Cheung 99] Tang, M., Eshraghian, K. and Hon Nin Cheung, H.N. A genetic algorithm for constrained via minimization, *Proceedings of the 6th International Conference on Neural Information Processing (ICONIP '99)*, Perth, WA, Australia, 16-20 Nov. 1999, vol.2, pp.435-440
- [Traci & Trevor 98] May-Plumlee Traci & Little, Trevor, No-interval Coherently Phased Product Development Model for Apparel, pp342-364, *International Journal of Clothing Science and Technology*, vol.10 no.5, 1998
- [Van der Aalst 99] Van der Aalst, W.M.P. Generic Workflow Models: How to Handle Dynamic Change and Capture Management Information. *Proceedings of the Fourth IFCIS International Conference on Cooperative Information Systems (CoopIS'99)*, Edinburgh, UK, 2-4 Sept. 1999, pp.115-126

[van der Aalst; Verbeek and Kumar 01] Van der Aalst, W.M.P. Verbeek, H.M.W. and Kumar, A. Verification of XRL: an XML-based workflow language, *The Sixth International Conference on Computer Supported Cooperative Work in Design*, London, Ont., Canada, 12-14 July 2001, pp.427-432

[W3C 98] Extensible Markup Language (XML) 1.0, World Wide Web Consortium Recommendation, Technical Report, 10-February-1998.  
(<http://www.w3.org/TR/1998/REC-xml-19980210>) 1998

[WfMC 96] Workflow Reference Model, Workflow Management Coalition (WfMC) 1996

[Xing, Wan, Rustogi & Singh 01] Xing, J. Wan, F. Rustogi, S.K. and Singh, M.P. Commitment-based interoperation for e-commerce, *Proceedings of 5th International Symposium on Autonomous Decentralized Systems*, Dallas, TX, USA, 26-28 March 2001, pp.161-168

[Yan and Chan 98] Yan, C.Y. and Chan, K.C.C. A Lotus Notes Implementation of a Workflow Automation Tool for ISO 9001 Certification, *Sixth International Conference on Software Quality Management (SQM'98)*, April 1998, pp.161-172

[Yan, Maamar and Shen 01] Yan, Y. Maamar, Z. and Shen, W. Integration of workflow and agent technology for business process management, *Proceedings of The Sixth International Conference on Computer Supported Cooperative Work in Design*, London, Ont., Canada, 12-14 July 2001, pp.420-426

[Zainudin and Hamdan 01] Zainudin, S. and Hamdan, A.R. A proposed design for a workflow engine, *Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology*, Singapore, 19-22 Aug. 2001, vol. 1, pp.65-68

[Zeng, Ngu, Bentallah & O'Dell 01] Zeng, L. Ngu, A. Bentallah, B. and O'Dell, M. An agent-based approach for supporting cross-enterprise workflows, *Proceedings of 12th Australasian Database Conference (ADC 2001)*, Australia, 2001, pp.123-130