MASTER

Product-based workflow design for cross-organization process monitoring
an ant colony optimization approach

Wang, T.

Award date:
2011

PRODUCT-BASED WORKFLOW DESIGN FOR CROSS-ORGANIZATION PROCESS MONITORING: AN ANT COLONY OPTIMIZATION APPROACH

Tingting Wang

Major of Business Information Systems

Eindhoven University of Technology

Supervisor: dr. M. (Marco) Comuzzi

Second Supervisor: dr.ir. I.T.P. (Irene) Vanderfeesten

Examination Committee: dr. D. (Dirk) Fahland

Eindhoven, July 2011

Preface

This master thesis is the outcome of my graduation project which completes my master study of Business Information System at the Eindhoven University of Technology. The project is conducted in Information System Group, Industrial Engineering and Innovation Sciences Department of the Eindhoven University of Technology.

The topic of the graduation project matches well with the courses in the master program and also fits my personal interests. Therefore, I enjoyed working on the project. I found the project challenging, as it required me to think outside knowledge from courses and come up with new ideas. Throughout the project, I have learned a lot of valuable things.

I would like to thank several people for their support during the project. First of all, I would like to thank my supervisor Marco Comuzzi for introducing me to such an interesting project and guiding me throughout the project. I would like to thank him for his trust, support, feedbacks, and pleasant cooperation. I also want to thank Irene Vanderfeesten for her suggestions and help. I would also like to thank Dirk Fahland being a member of my defense committee.

Special thanks to my parents and my boyfriend for their continuous support, trust and guidance. At last, many thanks to all of my relatives, friends, and other people that I cannot mention in detail for the support they gave. I would like to dedicate this thesis to all of you.

Tingting Wang
Eindhoven
July 2011

Abstract

Monitoring of processes in business networks requires the definition and implementation of monitoring processes that can deliver the right information to the right party in the collaboration.

Under such research background, this thesis proposes a method to (*i*) Capture monitoring requirements in cross-organization processes and (*ii*) Automatically derive the infrastructure to deliver correct monitoring information to the right party in the business network.

We propose a methodology based on PBWD (Product-Based Workflow Design) to solve the problem in our research. In particular, PBWD has been extended to fit our problem to include non-functional characterization of monitoring requirements e.g. cost, quality and availability information. Introducing non-functional characterization on operations in monitoring processes, we need to make the optimal choice available based on requirements from monitoring point of view. In this thesis we adapt ACO (ant colony optimization) to optimize the monitoring process obtained through PBWD. The implementation of our method is realized as plug-ins of the ProM framework.

Keywords: Monitoring, Product Data Model, Ant Colony Optimization

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This master thesis is the result of the graduation project for study Master of Science: Business Information Systems at the Technical University of Eindhoven (TU/e). The project is carried out at the Information Systems (IS) group of the Department of Industrial Engineering and Innovation Sciences Department at TU/e. In this thesis, we investigate an approach for optimizing Product data models (PDM) of cross-organization process monitoring with ant colony optimization (ACO). The approach presented here has been implemented in ProM by development of a set of plug-ins.

In Section 1.1, the context and motivation of this master thesis is explained. The research methodology is presented in Section 1.2. Section 1.3 provides the outline of the thesis.

1.1 Context and Motivation

Continuous monitoring of a business process can be defined as the set of methodology and tools to collect and disseminate relevant information about the process execution to interested stakeholders simultaneously with, or within a reasonably short period after, the occurrence of relevant events in the process [8]. Continuous monitoring has straightforward benefits, such as the opportunity for process providers to detect anomalies in (almost) real time and apply control actions on-the-fly [9].

In the scope of research in this thesis, the business network collaboration process is abstracted as PDM model. Product data models (PDM) is very suitable to express available information products and their dependencies, can be enriched with information about the cost of producing an information product or its quality for the interested stakeholders. With all these parameters, from monitoring point of view, we can choose a optimal pathway(combination of several process operations that can produce the end information product) based on the requirement of the stakeholders.

1.2 Research Methodology

In this thesis, we propose a methodology developed using ACO algorithm to design monitoring processes in collaborative business networks. The methodology considers as input the monitoring information made available by the collaborating parties and builds monitoring processes as PDM model. Moreover, monitoring information in our methodology can be described also in non-functional terms, e.g. by cost, quality, and availability. We adapt ant colony optimization algorithm in the problem to select optimal pathway of the monitoring process. The proposed methodology allows the selection of the monitoring process that satisfies also the party non-functional requirements, e.g. selecting the minimum cost monitoring process or the highest quality process, given a budget constraint.

1.3 Thesis Outline

The thesis is organized as follows. The citation review and research background are presented in Chapter 2, while Chapter 3 introduces the problem definition of optimal monitoring processes with PDM. The methods for optimizing monitoring PDM and, in particular, the ACO algorithm, are discussed in Chapter 4. Chapter 5 gives an evaluation of the ACO algorithm, using also a complex reallife PDM model. The implementation and execution of the approach in ProM are shown in Chapter 6, while conclusions are eventually drawn in Chapter 7.

Chapter 2

Research Background

In this chapter the research background of this thesis is introduced. First, the basic introduction about business networks is discussed in section 2.1. And then, the existing monitoring cross-organizational process methods are described based on different reference papers in section 2.2. Then the related work about business network modeling language: product data models (PDM) is given in section 2.3. At last, the previous work about application of PDM on monitoring cross-organizational process is discussed in section 2.4.

2.1 Business Networks

Faster market dynamics and fiercer competition push organizations to engage in complex, Internet- enabled, highly dynamic collaborations, referred to as collaborative Business Networks (BNs)[15]. Collaboration entails the enactment of cross-organizational business processes, which are regulated by agreements between the actors constituting the business network[7]. Typical examples of business networks can be found in the financial industry where, for instance, an insurance company may outsource parts of its process, such as loss adjustment, complaints management, and fraud detection, to external companies, in order to focus on its core business.

Information technologies (IT) supporting BNs help the network partners to lower coordination costs, i.e. reducing costs for communication, increasing opportunities for matchmaking, and enabling a more agile switch among business partners[22]. As a form of market-based collaboration, however, BNs, when compared with in-house business process execution, are characterized by higher control costs, which IT can help to reduce as well. When part of a business process is outsourced to external business partners, in fact, an organization faces the need for controlling the execution of the outsourced process. Control is characterized by risks derived from the possible inability of external providers to meet the expectations of the outsourcing organizations. Risks may translate into costs for the outsourcing organization. Control costs can be ascribed either to the opportunistic behavior of the external providers (moral hazard) or to their unsuitability to perform the task that has been assigned to them (adverse selection)[5].

2.2 Monitoring of Cross-organizational Processes

Cross-organizational workflow monitoring is based on a dynamic service consumer/provider paradigm[13]. This means that an organization that wants a service to be performed on its behalf (the service consumer) outsources this service to an organization that can perform this service (the service provider). This outsourcing is performed dynamically, which means that the decision to outsource is taken during the execution of the process requiring the service and that the provider is chosen dynamically. And all these interaction have to be monitored to help the collaboration for both parties.

Figure 2.1 illustrates the cross-organization collaboration approach. In this figure, we see how the service consumer outsources its activities D and E to a service provider that can perform these activities with an additional value (hence $D+$ and $E+$). The contract is the basis for the cooperation that, apart from service invocation and result reception, also encompasses detailed monitoring and control of the outsourced activities.

Monitoring information system can be mainly divided into two types. Typical $information-centric$ monitoring systems are composed of three layers: a layer that absorbs the stream of events coming from the business process execution engine, a processing and filtering layer that selects relevant events/data and automatically triggers actions, and a dashboard that allows users to follow

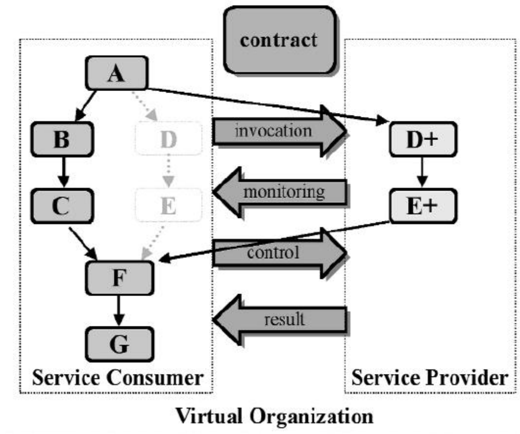

Figure 2.1: Dynamic service outsourcing in a virtual organization[13]

the process progress, view custom reports and statistics on the processes and send alerts. The events can also be logged and be available to be queried/mined posteriori[17]. The $process - centric$ monitoring system pay more attention on the process execution and run-time monitoring function of the system. As often alternative service providers with the similar or same functional but different performance characteristics are available from which the most suitable can be chosen, $process-centric$ monitoring system is more suitable than $information-centric$ systems for dynamic outsoursing in cross-organization monitoring.

Based on the importance of monitoring on cross-organizational processes, we searched the existing literatures on this topic and include some of the most important research papers here. In the following subsections, several monitoring methodology of cross-organizational processes are introduced. In subsection 2.2.1, an event based, non-intrusive monitoring framework for web service based systems in introduced[25]; In subsection 2.2.2, A pilot implementation of a continuous auditing system at Siemens is discussed[3]; In subsection 2.2.3, an approach of cross-organizational workflow management in dynamic virtual enterprises is provided[13].

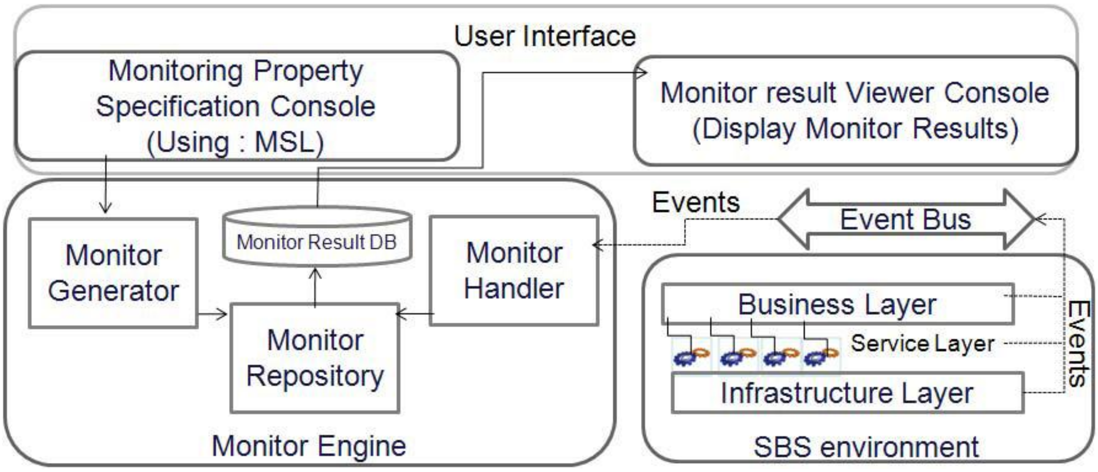

Figure 2.2: SBS monitoring Framework[25]

2.2.1 An Event Based, Non-Intrusive Monitoring Framework for Web Service Based Systems

A monitoring framework has been developed (as shown in Figure 2.2), which is independent of any business logic and service composition platform in this paper[25]. The monitoring engine works in parallel with the "Web Service Based Systems" (SBS) and allows for easy adaption of the business process. The SBS sends interesting events from the business layer, service layer (incoming/outgoing messages to/from the services used in the SBS) as well as from the infrastructure layer and feeds those into an event bus at runtime. A monitor observes the events from the event bus and accordingly monitors the functional and non-functional service composition assumptions and requirements of the SBS.

A Monitor Specification Language (MSL) has been developed to specify the properties of the system to be monitored at run-time. The language has the ability to specify boolean, statistical, and time-related properties. The specifications are automatically translated into executable C programs which act as run-time monitors. The monitors thus generated monitor the specified properties by capturing run-time events from the business layer, service layer and infrastructure layer of the SBS.

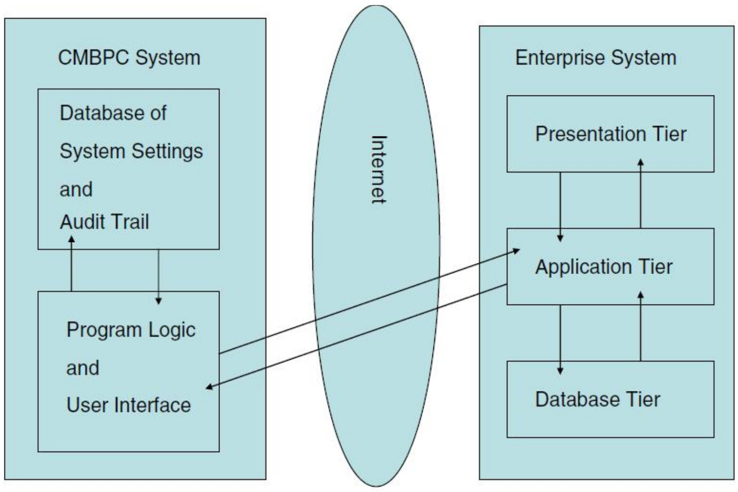

Figure 2.3: Architecture of the generic CMBPC system[3]

2.2.2 Continuous Monitoring of Business Process Controls: a Pilot Implementation of a Continuous Auditing System at Siemens

This paper introduces a continuous monitoring system in a real business environment inside Siemens[3]. The approach in this paper is a typical $information-$ *centric* approach. The architecture developed by the authors implements a completely independent continuous monitoring of business process controls (CMBPC) system running on top of Siemens' own enterprise information system which has read-only interaction with the application tier of the enterprise system as in Figure 2.3.

Among the key conclusions is that "formalizability" of audit procedures and audit judgment is grossly underestimated. Additionally, while cost savings and expedience force the implementation to closely follow the existing and approved internal audit program, a certain level of reengineering of audit processes is inevitable due to the necessity to separate formalizable and non-formalizable parts of the program. Our study identifies the management of audit alarms and the prevention of the alarm floods as critical tasks in the CMBPC implementation process.

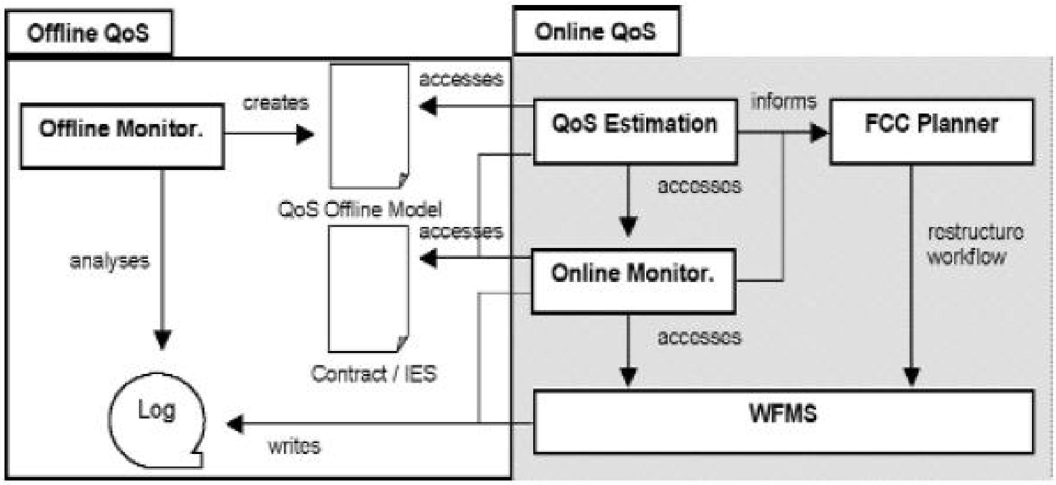

Figure 2.4: Interaction of QoS components[13]

2.2.3 CrossFlow: Cross-organizational Workflow Management in Dynamic Virtual Enterprises

This paper introduces the general architecture and technique of cross-organizational workflow management system[13]. It provides an approach of typical $process-centric$ system. Considering monitoring, the authors summarize the interplay between the cooperation support services components (SCC) for quality management in Figure 2.4. The quality of service (QoS) estimation component provides predictions of behavior of the currently running workflow instances. These estimates are based on performance models given as continuous time Markov models and produced by the offline monitoring component, which analyzes past executions of workflows, and on the behavior observed by the online monitoring component. The information on the current state of the workflow in conjunction with the performance model allows calculating probabilities for the future fulfillment of QoS parameters. In case the QoS estimation detects significant delays or deviations, the planning component of the FCC CSS is informed. It then has different options to restructure the workflow in order to achieve the most important QoS goals. The restructured workflow is then executed by the underlying workflow management systems.

2.3 Product Data Models

The product data mode (PDM) and product-based workflow design are firstly introduced in the manufacturing process in [23]. In manufacturing, the interaction between the design of a product and the process to manufacture this product is studied in detail. Consider, for example, material requirements planning (MRP) as part of current enterprise resource planning (ERP) systems, which is mainly driven by the bill of material (BOM)[20]. For information intensive products such as insurances, and many other services, the workflow process typically evolves or is redesigned without careful consideration of the structure and characteristics of the product.

PDM model is used in the analysis phase. When constructing a PDM model, all distinguished material that classifies as product specification is analyzed to identify the data elements, their dependencies, and the logic involved. The authors use a structure for administrative products that is somewhat similar to the BOM we discussed in the introduction. The BOM used in manufacturing is a tree-like structure with the end product as root and raw materials and purchased products as leafs. In the resulting graph, the nodes correspond to products, that is, end products, raw materials, purchased products, and subassemblies. The edges are used to specify composition relations (i.e., is-part-of relations). The edges have a cardinality to indicate the number of products needed.

In [23], a product data model is a tuple $(D, top, C, pre, F, constr, cst, flow, prob)$ [23]:

- D: a set of data elements, with a special top element top:

$$top \in D,$$

- C: set of constraints; a constraint can be any Boolean function; the function that always yields truedenoted trueis part of C:

$$true \in C,$$

- the function pre gives for each information element the various ways of determining a value for it on basis of the values of different sets of other information elements:

$$pre : D \rightarrow \mathcal{P}(\mathcal{P}(D)) \text{ such that,}$$

- F: a set of production rules, based on the definition of *pre*; F consists of all ordered pairs of data elements between which a dependency may exist:

$$F = \{(p, cs) \in D \times \mathcal{P}(D) | cs \in pre(p)\},$$

- the function *constr* that associates a constraint to each production rule:

$$constr : F \rightarrow C, \text{such that}$$

- a function *cst*, which gives the cost of using a production rule:

$$cst : F \rightarrow \mathbb{N},$$

- a function *flow*, which gives the time it takes to use a production rule:

$$flow : F \rightarrow \mathbb{N},$$

- a function *prob*, which gives the probability that a production rule will yield an acceptable result when used:

$$prob : F \rightarrow (0..1], \text{ such that}$$

Figure 2.5 shows a example helicopter pilot product data model based on the formalization. All boxes in this figure correspond to data elements. Arcs are used to express the pre relation that is in use to decide whether a candidate is suitable to become a helicopter pilot in the Dutch Air Force. Multiple starts of an arrow are joined into one by small black dots. For example, the outgoing arrows of

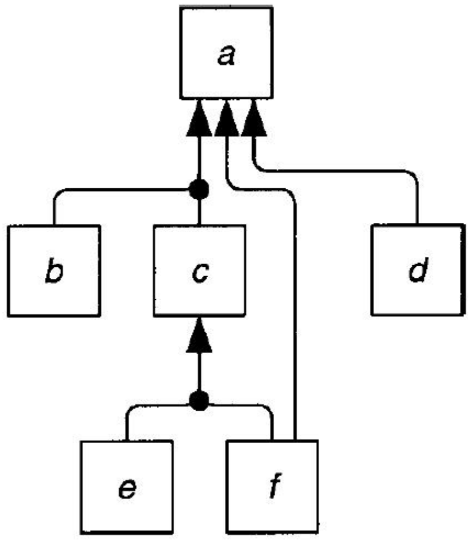

Figure 2.5: Helicopter pilot product data model[23]

data elements b and c are joined into one single arrow leading to data element *a*. It represents a production rule for a with data elements *b* and *c* as inputs. The meaning of the data elements is as follows:

- a: suitability to become a helicopter pilot
- b: psychological fitness
- c: physical fitness
- d: latest result of suitability test in the previous two years
- e: quality of reflexes
- f: quality of eyesight

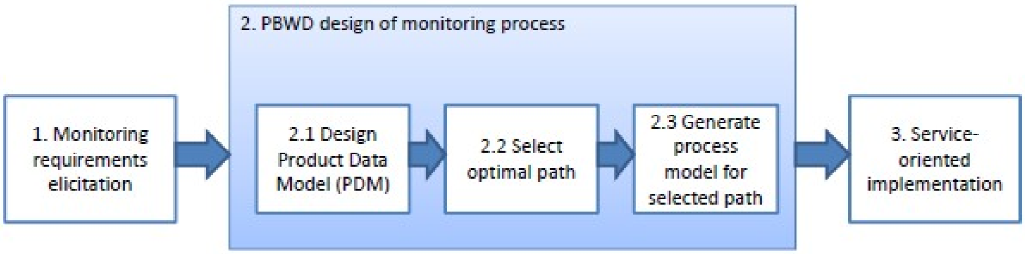

Figure 2.6: Methodology for PBWD of monitoring processes[9]

2.4 Application of PDM on Monitoring Cross-organizational Processes

The methodology and possibility to use PDM model in monitoring cross-organizational processes is discussed in [9]. While the information-centric view can suffice for intra-organizational process monitoring, where all monitoring information is produced in a given business domain, in cross-organizational settings researchers stress the importance of process and communication-oriented mechanisms to transmit relevant information to interested parties across the collaborative network. In other words, once the monitoring information is captured and made available by the collaborating parties, a process must be built to allow a specific party to retrieve (or be delivered) the monitoring information in the right way. Such monitoring process should account for the temporal and aggregation dependencies among monitoring information.

In this paper, the authors propose a methodology to design monitoring processes in collaborative business settings. The methodology considers as input the monitoring information made available by the collaborating parties and builds monitoring processes embedding temporal and aggregation dependencies among monitoring information. Moreover, monitoring information in this methodology can be described also in non-functional terms, e.g. by cost, quality, and availability. The proposed methodology allows the selection of a monitoring process that satisfies also the party non-functional requirements, e.g. selecting the minimum cost monitoring process or the highest quality process, given a budget constraint.

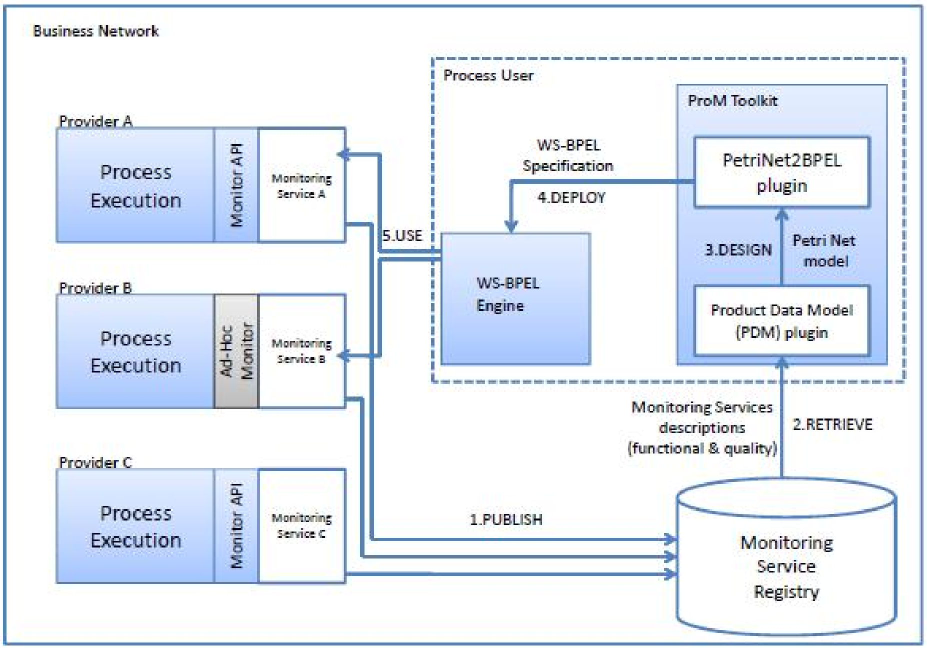

Figure 2.7: PBWD-based monitoring process creation architecture[9]

Figure 2.6 shows the methodology for PBWD of monitoring process in this paper. The Service oriented implementation architecture for this paper is shown in Figure 2.7.

In general, what we focus in the research of this thesis is on monitoring the process information in business network by applying ant colony algorithm optimization on PDM model. Different from the previous mentioned paper, in our research there are two main innovation points. In [9], the algorithm of optimization the optimal pathway is not defined. In our research, we adapt ant colony optimization in finding the optimal pathway in the PDM model. Also, we realize the implementation of both the optimization in ProM 5.2 and the execution of the optimization result based on online interaction with the client. These implementation realize the evaluation of the methodology and can be seen as a proof of our proposed methodology.

Chapter 3

Problem Definition of Optimal Monitoring Processes with PBWD

In this chapter the research problem of this project is described and formalized. Also, relevant knowledge and expression examples are provided. First, the basic introduction about optimization problem is described in section 3.1. Then the formalization of definitions in PDM model is given with examples in section 3.2. After that, a case study is provided in section 3.3. At last, the optimization requirement is discussed with explanation on the case study in section 3.4.

3.1 Problem Conceptional Description

The PBWD methodology takes the structure of the informational product, which is described in a Product Data Model (PDM), as a starting point to derive a process model. The structure of a workflow product is captured by a Product Data Model [23] in our research. The PDM is described by a tree-like structure [31]. An informational product is, for instance, a decision on an insurance claim, an allocation of a subsidy, or an approval of a loan, and so on. Based on the input data provided by the client or retrieved from other systems, the end (informational) product is constructed step-by-step. In each step, a new information

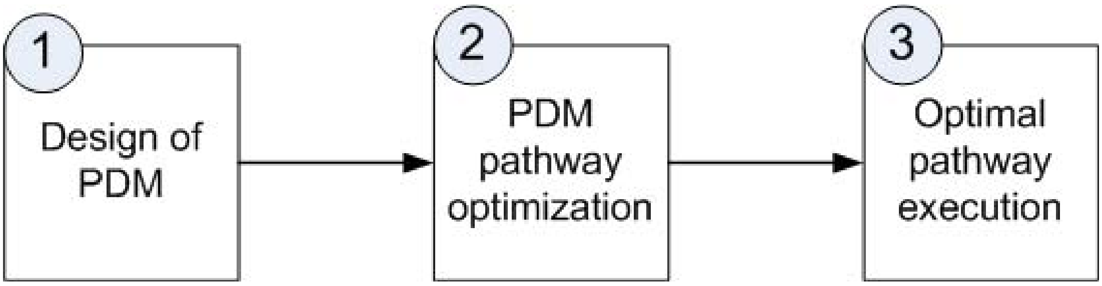

Figure 3.1: The research procedures

is produced based on the specific data available for the case [9].

As shown in Figure 3.1, the first step of the research procedure is design of the PDM model that capture the monitoring requirements of the stakeholders. There are several methods to construct the PDM model of the business process, such as retrieving data from the datalogs, analysis of the literal material or brainstorm with the expert of the process. But this part is out of the scope of this thesis. The next step of this approach is the selection of an optimal pathway in the PDM. The PDM model may accommodate several alternative pathways to produce the end product. During the selection of the optimal pathway, different constraints may be considered. The objective of this step is therefore to select a complete pathway i.e. a pathway containing the root element of the PDM, which satisfies the requirements of the considered stakeholder. In the research of this thesis, we consider cost, quality, or availability of the monitoring information.

In this research, we abstract the parameters for each operation as Cost (c), Quality (q) and Availability (a). These are not the only parameters influencing the business process, there may be other parameters that stakeholder cares, such as throughput time and so on. For the clarity, we only take into consideration the three above in this project and consider them unified together instead of separately. The proposed methodology allows the selection of a monitoring process that satisfies the party non-functional requirements, e.g. selecting the minimum cost monitoring process, or the highest quality process, given a budget constraint [9].

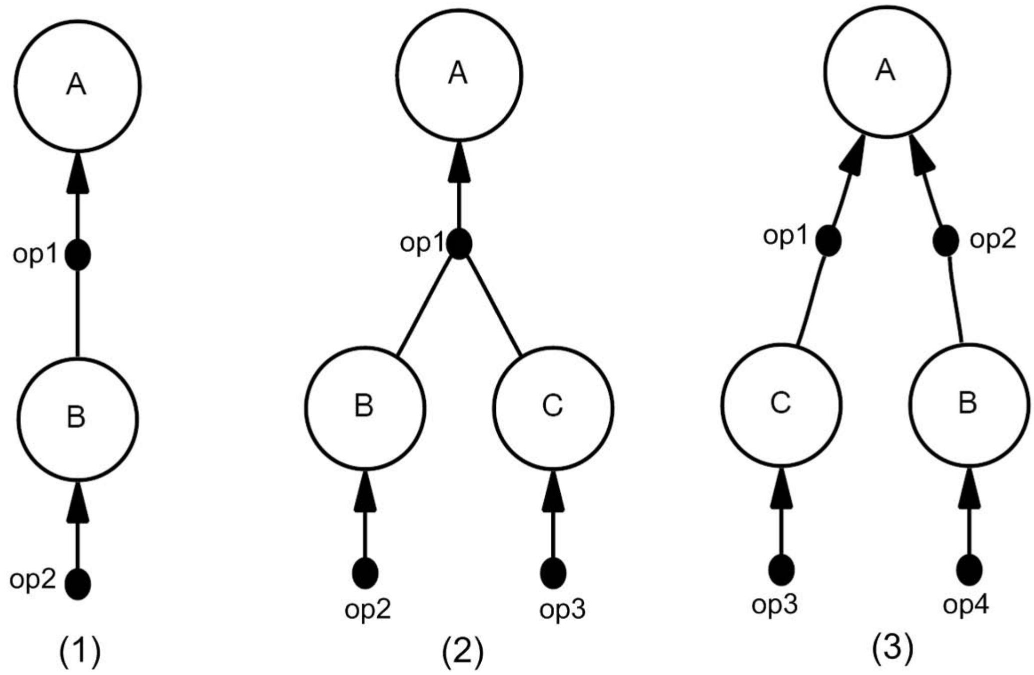

Figure 3.2: Basic PDM model examples

3.2 Formalization of PDM

Definition 1 (Information data element): Let D be a set of information data elements. An element of D is represented as a circle in the PDM data models as shown in Figure 3.2. In PDM model (1), $D = \{A, B\}$, A is the root element *root* which is included in the set D.

The actions that can be taken on the data elements are called operations and are represented by (hyper) arcs in the PDM. Each operation may have zero or more input data elements and produces exactly one output data element. The arcs are "knotted" together when values for a set of data elements are needed to execute the particular operation. Formally, the operation is defined as follows:

Definition 2 (Operation): Let OP be a set of operations. An element of OP is represented as a arrow with a dot in the PDM data model as shown in Figure 3.2. An element of OP, that is a particular operation, is denoted as $op = (in, out)$, where

- $in \subseteq \mathbb{P}(D)$ is the set of input data elements;

- $out \in D$ is the output data element;
- $in \bigcap out = \emptyset$.

For simplicity, we let $op.in$ and $op.out$ represent the input data elements and output data elements of a particular operation op, respectively. As shown in PDM model (1) of Figure 3.2, $OP = \{op1, op2\}$, $op2.out = B$, $op1.in = B$ and $op1.out = A$. Note that the input data elements in of a particular operation could be $\emptyset$. For example, the input data elements of the operation op2 is empty, as shown in Figure 3.2 model (1).

Also, for each operation, we consider three parameters in this project:

- $v_q(op_i)$ is the quality of the operation op_i, $v_q(op_i) \in [0, 1]$, $\forall op_i$;
- $v_c(op_i)$ is the cost of the operationop_i, $v_c(op_i) \geq 0$, $\forall op_i$;
- $v_a(op_i)$ is the availability of the operation op_i, $v_a(op_i) \in [0, 1]$, $\forall op_i$;

Definition 3 (PDM): A product data model is a tuple $\{D, OP, F\}$ which consist of information data elements D, operations OP and the production rule F between them.

$$F : (\mathbb{P}(D) \times OP \longmapsto D) \tag{3.1}$$

In PDM model (1) of Figure 3.2, we can see that there are two production rules in the model: $F(\phi, op2) = B$ and $F(B, op1) = A$.

In the PDM model, there are different kinds of production rules which represents the relationship between data elements and operations. Based on the connection of operations and production rules, the logic between input and output data elements can be generally divided into three types.

- If a particular data element $d \in D$ is the output data element for only one operation, and this operation has zero or one input data element, we say that there is a $STRAIGHT$ logic between the input element (or no data element) and the output data element. The examples are $op1$ and $op2$ in PDM model (1) of Figure 3.2.
- If a particular data element $d \in D$ is the output data element for only one operation, and this operation has multiple input data elements, then this is

a AND logic between the input elements and the output data element.The example is $op1$ in PDM model (2) of Figure 3.2.

- If a particular data element $d \in D$ is the output for multiple operations, then we need to choose among one of them based on the parameters mentioned above, this is a OR logic between the input elements and the output data element.The examples are $op1$ and $op2$ in PDM model (3) of Figure 3.2.

After defining the structure of the PDM, the notion of pathway and complete pathway need to be formalized in the model.

Definition 4 (PDM operation pathway): A pathway is a set of operations in the PDM that are consecutive. Let $Path \subseteq \mathbb{P}(OP)$ be the set of all the pathways in the PDM model. A pathway is all the operations between data elements that are connected by the three logic types from the root data element (as the first output data element). For example, as shown in Figure 3.2, the three models have different pathway construction from root element "A": In model (1), A will be connected to B by op1, either $\{op1\}$ or $\{op1, op2\}$ can be pathways in the model; In model (2), A will be connected to both B and C by op1, $\{op1, op2\}$ can be pathway in the model; In model (3), A will be connected to either B by $op1$ or to C by op3,$\{op1, op3\}$ can be pathway in the model.

Definition 5 (Complete pathway): A ***complete*** pathway is a subset of operations in the PDM model that leads to the production of the root element of the PDM starting from a set of leaf operations.

In particular, we denoted the set of complete pathways as P. A PDM pathway is a ***complete*** pathway if the following conditions are satisfied:

1. Starting from the root data element, only stop including operations when there is no input data element left, i.e. we reached a (set of) leaf operations in the PDM.

2. There are no other operations included which have same output element with the operations which belong to the pathway, i.e., $\forall op'$, if $\exists op \in P$, $op.out \cap op'.in \neq \emptyset$, then $op' \in P$.

For example, the pathway $\{op1, op2\}$ is not a complete pathway in PDM model (2) as shown in Figure 3.2 based on the first rule. Only pathway $\{op1, op2, op3\}$

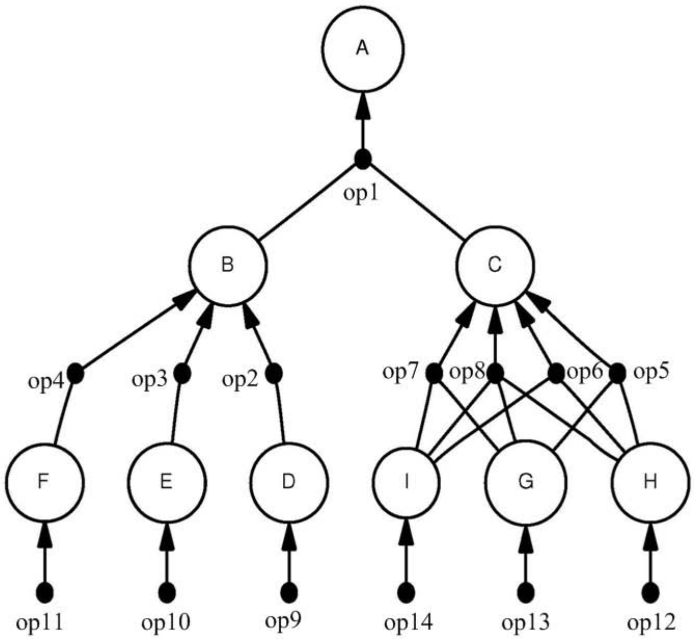

Figure 3.3: The Product Data Model of loss evaluation

is a complete pathway in model (2). The pathway $\{op1, op2, op4\}$ is also not a complete pathway in PDM model (3) based on the second rule because it as a "dummy" operation $op4$. In model (3), either pathway $\{op1, op3\}$ or $\{op2, op4\}$ can be complete pathways from the model.

3.3 Case Study Explanation

The example, shown in Figure 3.3 with explanation of each data element in Table 3.1, is a typical expression of a simple PDM needed to be optimized in order to find the optimal complete pathway [8]. In this example, an insurance company may define an offer for its prospective customers referring to the damage loss evaluation. Our case study model is about how to monitor this process, cov-

Table 3.1: The product data elements of loss evaluation

data element	information product statues
A	Damage loss evaluation
B	Fraud check auditor evaluation
C	Loss adjustment evaluation
D	Fraud check auditor supplier 1
E	Fraud check auditor supplier 2
F	Fraud check auditor supplier 3
G	Loss adjustment claim evaluation 1
H	Loss adjustment claim evaluation 2
I	Loss adjustment claim evaluation 3

ering the status of the information products and the cost, quality and availability parameters to get the monitoring information. The general evaluation steps has the following structure: the loss adjustment activity (data element C) is outsourced to a risk manager, the fraud check (data element B) is outsourced to an external 3rd party auditor, and the company makes some internal processing to integrate the results of these two activities to product the end output damage loss evaluation (data element A).

It is supposed (for the sake of clarity rather than reality) that there are just three suppliers for each insurance checking activities. The status of output result from each supplier is abstracted to data element D-I. The parameters to get the monitoring information for each operation are shown in Table 3.2.

In this case study, the definitions we mentioned above can be figured out. Figure 3.3 is the PDM model expression of this case study. Based on the different logic inside the model, we can get a pathway like $\{op1, op4, op7\}$ from root element A, which is not a complete pathway because it does not satisfy the first condition of complete pathway. In the example product data model, the pathway $\{op9, op2, op12, op13, op14, op08, op1\}$ on the example PDM is a complete pathway, since it has and only has all the data element which leads from the root element A in the PDM to all the operations that connected.

Table 3.2: Loss evaluation operation parameters.

Operation	v_c	v_q	v_a	Operation	v_c	v_q	v_a
Op01	0.0	1.0	1.0	Op08	0.9	0.9	1.0
Op02	0.8	0.9	1.0	Op09	1.0	1.0	0.8
Op03	0.2	0.7	1.0	Op10	0.2	1.0	0.4
Op04	0.6	0.8	1.0	Op11	0.9	1.0	0.7
Op05	0.2	0.6	1.0	Op12	0.5	1.0	0.6
Op06	0.6	0.5	1.0	Op13	0.8	1.0	0.9
Op07	0.7	0.8	1.0	Op14	0.3	1.0	0.9

3.4 Optimization Requirements

After formalizing the ***complete*** pathway and parameters on each operation, the third step of this research, as shown in Figure 3.1, would be defining the optimization methodology for choosing the best complete pathway in the monitoring PDM model. In the general case, the selection of an optimal pathway should be seen as an optimization of a utility function. The utility function is defined by the monitoring stakeholder, as a weighted sum of partial utility functions on individual dimensions[4]. The partial utility $V_q(p)$, $V_a(p)$ and $V_c(p)$ which represents the quality, availability and cost of the whole complete pathway in PDM model are defined as follows:

$$V_q(P) = \min_{op \in P} v_q(op_i)$$

$$V_a(P) = \prod_{op \in P} v_a(op_i)$$

$$V_c(P) = \sum_{op \in P} v_c(op_i)$$

More complex utility functions, e.g. the indivisible functions which influence each other, are out of the scope in this research. In this way, we can define the utility for a complete pathway as:

$$U(P) = w_q * V_q(P) + w_a * V_a(P) - w_c * V_c(P) \tag{3.2}$$

For clear expression, $U(p)$ can be also expressed as:

$$U(P) = w_q * \min_{op \in P} v_q(op_i) + w_a * \prod_{op \in P} v_a(op_i) - w_c * \sum_{op \in P} v_c(op_i) \tag{3.3}$$

Where $w_q + w_a + w_c = 1$. There are two types of problems in our study as explained in the following.

Problem 1: The first kind of problem is choosing an optimal pathway given the fixed weights for these three parameters without other constraints. Thus, an optimal pathway should have the maximum utility:

$$U(P) \to Maximum \tag{3.4}$$

Where P is a complete pathway on the target PDM.

Problem 2:Another more complex requirement from the stakeholder may be the optimization of one parameter given side constraints on the others. Then the optimization result will be influenced by the constraints. The constraints that the optimization algorithm should subject to can be expressed as:

$$V_q(P) = \min_{op \in P} v_q(op_i) \geq \varphi_q \tag{3.5}$$

$$V_a(P) = \prod_{op \in P} v_a(op_i) \geq \varphi_a \tag{3.6}$$

$$V_c(P) = \sum_{op \in P} v_c(op_i) \leq \varphi_c \tag{3.7}$$

where p is a pathway, and $\varphi_a \in [0, 1]$ is the threshold value of availability, $\varphi_c \in \mathbb{N}$ is the threshold value of cost, and $\varphi_q \in [0, 1]$ is the threshold value of quality. The pathway which satisfies the above Equations 4-6 is then the set of optimal resource allocations for the stakeholder.

When considering the case study example, from the PDM shown in Figure 3.3, and the operational properties, shown in Table 3.2, we can derive 12 complete pathways for the monitoring process that may all serve the need for monitoring the insurance claim evaluation process. These pathways are reported in Table 3.3 with their properties for the whole complete pathway. Based on the information

Table 3.3: Loss evaluation pathways

Pathway	Operations	V_c	V_q	V_a
1	Op1 Op2 Op9 Op5 Op12 Op13	3.3	0.6	0.432
2	Op1 Op2 Op9 Op6 Op12 Op14	3.2	0.5	0.432
3	Op1 Op2 Op9 Op7 Op13 Op14	3.6	0.8	0.648
4	Op1 Op2 Op9 Op8 Op12 Op13 Op14	4.3	0.9	0.3888
5	Op1 Op3 Op10 Op5 Op12 Op13	1.9	0.6	0.216
6	Op1 Op3 Op10 Op6 Op12 Op14	1.8	0.5	0.216
7	Op1 Op3 Op10 Op7 Op13 Op14	2.2	0.7	0.324
8	Op1 Op3 Op10 Op8 Op12 Op13 Op14	2.9	0.7	0.1944
9	Op1 Op4 Op11 Op5 Op12 Op13	3	0.6	0.378
10	Op1 Op4 Op11 Op6 Op12 Op14	2.9	0.5	0.378
11	Op1 Op4 Op11 Op7 Op13 Op14	3.3	0.8	0.2835
12	Op1 Op4 Op11 Op8 Op12 Op13 Op14	4	0.8	0.3402

provided in Table 3.2, selecting the highest quality process, given a budget constraint would not be hard as all the pathways and parameters are listed clearly. But for the real life PDM models, which would be much more complex, it is much harder to evaluate all the complete pathways and all the parameters. This is the reason we need to develop a methodology to search an optimal pathway on the target PDM in order to suit the requirements for the monitoring process of the monitoring stakeholder.

Chapter 4

Methods for Optimal Monitoring Processes using PDM

After defining the optimization problem, the next step is discussing and testing different methods to solve this problem. In this chapter the methods that can be used on this optimization problem are discussed. First, the exhaustive search methodology is described in section 4.1. Then the local greedy optimization is given with test of its benefit and limitations in section 4.2. At last, the ant colony optimization algorithm is adapted on this study and tested with case study example in section 4.3.

4.1 Exhaustive Search

To begin with, the enumeration algorithm is the first to be analyzed. As discussed in section 1.4, enumeration can be used to find an optimal pathway on the target PDM. However, this method requires to enumerate all possible pathways in the PDM and to calculate the utilities separately as in the case study section. The main problem we may confront is the logic to keep memory of which branch has been calculate already until now. Although this method always leads to the selection of the optimial pathway, there is duplicate calculation in this algorithm which increase both time and space complexity in execution. Therefore, this method is only suitable to compare with other methodologies, to test if they manage to find the optimal pathway. It is also explained in the experiment section how complex it can be to list all the pathways and calculate all the parameters for a middle size real life business network PDM model.

4.2 Local Greedy Optimization

The simple local greedy algorithm is also considered in this problem because it is faster than many other algorithms based on its inside logic. In order to choose a complete pathway, starting from the root element of the target PDM model, we should connect any $STRAIGHT$ or AND input data element of the operation. If a data element is connected to multiple OR logic input elements, the best one i.e. the one with maximum utility is chosen based on the local comparison of operations by which input data elements are connected. The general optimization method is expressed in algorithm 1.

We can simply explain the algorithm on the basic PDM model examples shown in Figure 4.1:

- For $STRAIGHT$ logic structure as shown in PDM model (1), begin from $(A, \emptyset)$ as shown in line 1,$op1$ is the only operation in op_list as shown in line 8. Therefore, $op1$ in included into P as shown in line 10 and "Search_Node" goes on. Now the Input is$(B, op1)$, there is still only one operation available. Finally, we get $P = \{op1, op2\}$ and complete the search.

- For AND logic structure as shown in PDM model (2), begin from $(A, \emptyset)$,$op1$ is the only operation in op_list. $op1$ in included into P and "Search_Node" goes on for both the two input data elements B and C. The next search Input is$(B, op1)$, there is only one operation available without input data element. So $op2$ is included in P and this branch is completed. Finally, the other branch is executed and we get $P = \{op1, op2, op3\}$ and complete the search.

- For OR logic structure as shown in PDM model (3), the case is different. Begin from $(A, \emptyset)$, there are two operations $op1$ and $op2$ in op_list. Then we need to calculate the utility using Eq. (3.3) as shown in line 9. Only one

Algorithm 1 Local strategy to search pathway

Procedure::Search_Pathway(*root*)
Input:
    root is the root data element of a PDM
Output:
    P is the optimal result pathway of a PDM
Steps:
    Let $P = \emptyset$
    Search_Node($root, P$)
    return P
End Procedure

Procedure::Search_Node(x, P)
Input:
    x is a particular data element of a PDM
    P is a partial result pathway of a PDM
Output:
    P is a partial result pathway of a PDM
Steps:
    Let $op\_list = x.in$ be the input operation list of x
    Select an operation *op* from *op_list*, where *op* has maximum $U(p)$ calculated from root element till *op* using Eq. (3.3).
    Let $P \leftarrow P \cup \{op\}$
    For each *node* in input data element lists of *op*
        Search_Node($node, P$)
    End For
End Procedure

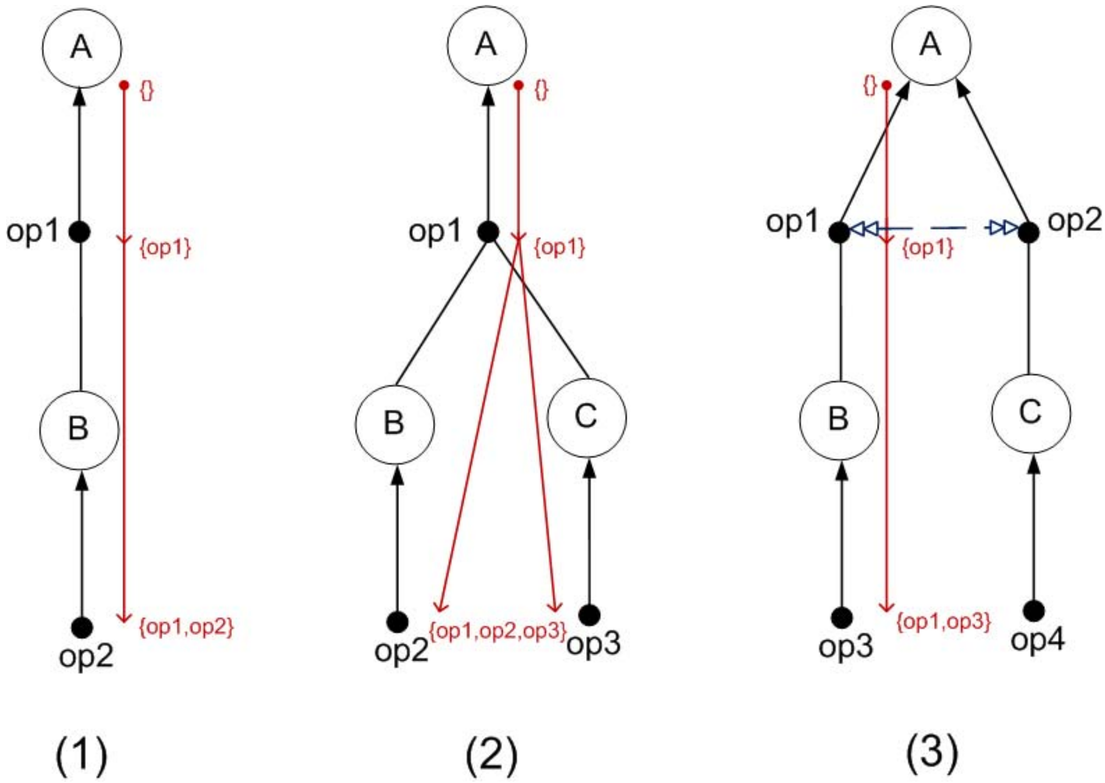

Figure 4.1: Local greedy algorithm on different logic models

operation will be chosen to be included in P. We suppose $op1$ is the one with lower utility calculated between $op1$ and $op2$ (as there is no preceding operation before them in the partial pathway). Then $op1$ is included and we goes on search until get $P = \{op1, op3\}$ as the search result.

Figure 4.2 shows a screenshot of a PDM plug-in implementing the local greedy algorithm. More details of the implementation are given in Chapter 5.

When considering time cost of the optimization, we can see from Figure 4.2 that local optimization on a small PDM model as the case study consume very little time (less than 1 millisecond). We can get the conclusion that scanning the model only once with local calculation is a very efficient method considering time complexity. However, as we will discussed, it is not optimal from the utility point of view.

This method cannot guarantee the selected pathway satisfies the predefined constraints with utility. This is because from the top-down sight of view, the optimal pathway can not be chosen only based on local optimization. As discussed in [14], greedy algorithms mostly (but not always) fail to find the globally optimal solution, because they usually do not operate exhaustively on all the data. They can make commitments to certain choices too early, preventing them from finding the best overall solution later.

Considering the case study, based on the calculation in Table 4.1, the highest availability pathway is pathway 3. The total availability of this plan is 0.648. However, if we use the greedy strategy to the PDM, we can find problem in choose among op2, op3, and op4 already. As the availability of all these three operations equal to 1, we can only randomly choosing one in our algorithm. Moreover, when the cost is optimized as in Figure 4.2, among $op5$, $op6$, $op7$, and $op8$, $op5$ (0.2) is definitely chosen as the cost optimized operation. But the total lowest cost pathway is actually pathway 6 with only 1.8. We can see from the small example that this strategy is not perfectly fitting our requirement.

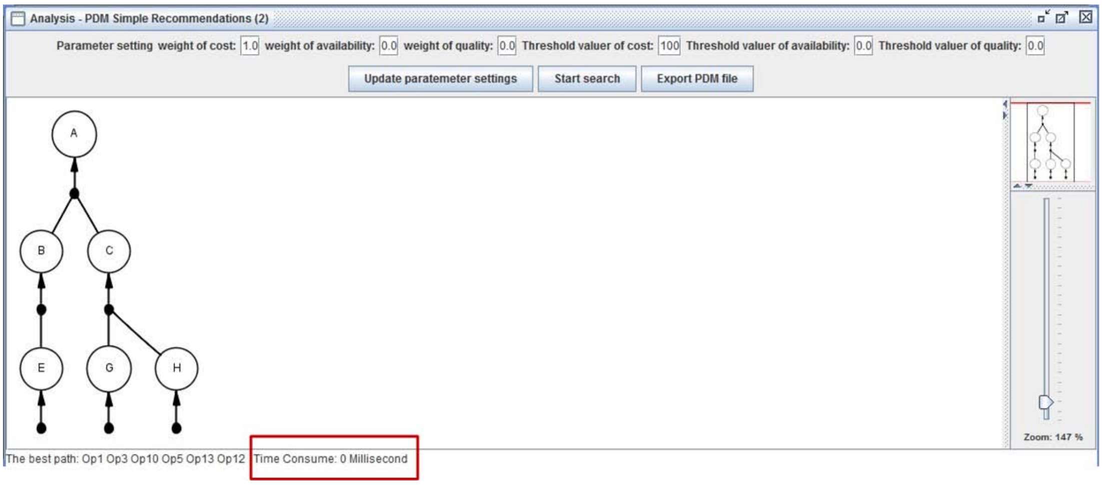

Figure 4.2: Local greedy optimization result on cost

4.3 Ant Colony Optimization Algorithm

4.3.1 Ant Colony Optimization

The last method we propose to obtain the optimal pathway in the PDM model is heuristic. A global decision strategy takes into account the effect of the current operation on future operations. The effect of selecting a particular operation on the future operations is considered when making a resource allocation decision for the current task. A global decision strategy takes the complete, alternative pathway into account to optimize the overall performance of the case.

Efficient and fast searching of the optimal pathway in the PDM, is very important in this application. Generally, searching an optimal pathway is a NP hard problem[16]. To the best of our knowledge, there is no correct efficient solution available. Thus, near optimal solutions with reduced computation becomes more important. In this research, we present an Ant Colony Optimization (ACO)-based strategy in order to search optimal pathway on the PDM.

ACO is a technique of problem solving inspired by the behavior of ants in finding paths from the nest to food and a new search metaphor for solving combinatorial optimization problems. ACO has been remarkably successful in recent years [10, 11, 12, 18, 19, 21]. As shown in Figure 4.3, in the natural world, ants (initially) wander randomly, and upon finding food return to their colony while laying down pheromone trails. If other ants find such a trail, they are likely not to keep traveling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food. Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromone have to evaporate. A short path, by comparison, gets marched over more frequently, and thus the pheromone density becomes higher on shorter paths than longer ones. Pheromone evaporation also has the advantage of avoiding the convergence to a locally optimal solution. If there was no evaporation at all, the paths chosen by the first ants would tend to be excessively attractive to the following ones. In that case, the exploration of the solution space would be constrained. Thus, when one ant finds a good (i.e., high utility in our case) path from the colony to a food source, other ants are more likely to follow that path, and positive feedback eventually leads all the ants following a single path. The idea of the ant colony algorithm is to mimic this behavior with "simulated ants" walking around the graph representing the problem to solve.

The basic idea of the ACO algorithms [10, 11, 12] is that ants choose the trail with largest concentration of pheromone τ with a certain probability $prob$. The optimal path will have more pheromone concentration. For a whole colony following this behavior the optimal path will be much higher in pheromone concentration than the other paths. Then the probability of choosing any other paths will be very small. Also the less used path will be much lower in pheromone concentration due to pheromone volatility with time [11, 12].

The reason for us to choose ant colony optimization among other heuristic algorithms in our research is its fitness with our requirements. We can see that the root data element can be used as the nest in the algorithm. The set of leaf operations can be used as the food for the ants. One ant search is similar to one complete pathway in PDM model. We can record the utility data for one complete pathway based on one ant search. Furthermore, we can set a number of ants as one iteration of learning process in the model. With the utility data gathered from one iteration, we can update the pheromone to optimize the chance to follow the pathway with higher utility. The only problem in our research that

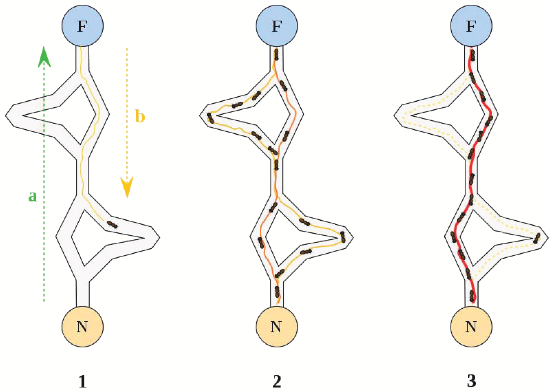

Figure 4.3: Ant colony optimization.

does not fit with classic ACO algorithm is how to deal with the AND logic with multiple input data elements in the PDM model. For this reason we extended the classical ACO algorithm as explained in the next section.

4.3.2 Adapt Ant Colony Optimization Algorithm to PDM Optimization

In this work, we develop a specific ACO-based algorithm to address the optimal operation pathway searching problem. The proposed algorithm is a variant of the traditional ACO algorithms. The description of the proposed algorithm is given in the following.

Based on the classic ACO as in [6], this project needs one extension to adapt the classic program to the requirements. As there is AND logic in the PDM model, to produce a data element may need more than one input data elements. There could be multiple branches from the AND logic operation. To cover all the input data elements and their following operations, when one ant goes through a pathway including AND operation, offspring of the ant is needed. The ant will be duplicated to cover all the input data elements of the operation. For example as shown in the case study example in Figure 4.1(2), when an ant path $op1$, it will become two child ants to data element B and C which records the utility information on both branches. The general algorithm for applying ACO on this problem is shown in algorithm 2.

The number of ants in each iteration, and the iteration number are adjustable in this study. For each iteration, an amount of ants go through the PDM model from the root data element. After each ant and all its child ants finish the journey, the utility information from the complete pathways they have went through are recorded on each ant. At the end of each iteration, the amount of pheromone deposited is then weighted for each solution, such that the solutions with better fitness deposit more pheromone than the solutions with worse fitness.

Initially, we put equal amount of pheromone trails on all operation nodes on the PDM. The purpose of this step is to give a random guide to the first ant to follow. In principle, we adopt backward search solution. Thus, we put the first ant at the root data element on the PDM. The ant chooses one of the available pathways guided by the pheromone trail, and goes to the next node. The node reached by the first ant is now considered as the initial state. Then the ant goes to the next node guided by the pheromone trail and so on until it reaches the final nodes on the PDM. After passing through a complete pathway, the utility is calculated and assigned as pheromone trace on this pathway at the end of this iteration. Note that there may exist multiple operations which connect to a particular data element node. The principle of selecting one particular operation node is given as follows:

In step 14 of algorithm 2, the idea of how to deposit suitable amount of pheromone is considered. Based on the classic Ant Colony Optimization method [6], let an ant *ant* reside in a node x at time t. Assume that one of the possible trails for its next step is $x- > op_i$ connecting node x to operation op_i. The probability of that ant selects this trail is formalized as:

$$p(x, op_i) = \frac{\tau(x, op_i)^\alpha}{\sum_{op_i \in x.in} \tau(x, op_i)^\alpha} \tag{4.1}$$

where $p(x, op_i)$ is the pheromone concentration on the operation node op_i. And the pheromone update function is given by:

$$\tau(x, op_i)(t+1) = \beta * \tau(x, op_i)(t) + \sum_{j=1}^{j=num} \psi(ant_j, x, op_i) \tag{4.2}$$

Algorithm 2 Using ant colony algorithm to search pathway

```
Procedure::ACOAlgorithm(root, Num, Iterations)
Input
    root: the root node of a PDM.
    num: the number of ants.
    iterations: the number of iterations.
Output
    bestPath = ∅: the optimal pathway on the PDM.
Steps
    For each iteration
        For each ant ant do
            let P = ∅ be a pathway of a PDM
            ACOSearchAlgorithm(root, ant, P)
            calculate utility U(p)of P according to Eq. (3.3).
            determine the amount of pheromone to be deposit on every arc on the P based on the utility of the P.
        End For
        perform a pheromone update using Eq. (4.2).
        If bestPath = ∅ or U(p) > U_bestPath, then
            bestPath = P
        End If
    End For
    return bestPath
End Procedure
```

```
Procedure::ACOSearchAlgorithm(node, ant, P)
Input
    node: a particular data element on a PDM.
    ant: a particular ant.
    P: a particular pathway on a PDM.
Output
    P = ∅: a partial pathway on a PDM.
Steps
        select the suitable operation op in the adjacent operations of node according to the transition rule using Eq. (4.1).
        add edge from node to op to the P
        For each node node' in the preceding nodes of op do
                add edge from op to node' to P
                ACOSearchAlgorithm(node', ant, P)
        End For
End Procedure
```

where $\beta \in (0,1)$ is the evaporation coefficient, num is the number of ants in each iteration, $\psi(ant, x, op_i)$ is the pheromone released by the ant k on the trail $x- > op_i$, ψ is the utility of the pathway P that ant ant (including its children) searches.

For the first kind of problem mentioned in section 3.4, $\psi(ant, x, op_i) = U(p)$ when the operation is in the pathway of this ant.

$$\psi(ant, x, op_i) = \begin{cases} U(p) & op_i \in p_{ant} \\ 0 & op_i \notin p_{ant} \end{cases} \tag{4.3}$$

where p_{ant} is the pathway that an ant ant searches, U(p) is the calculated utility of that pathway based on Equation 2.

For the second kind of problem with side constraints from stakeholder in section 3.4, the general algorithm and probability calculation equation are the same. The way to calculate pheromone $\psi(ant, x, op_i)$ is different by adding the side constraints. If a pathway violates the constraints shown in Eq. (3.5-3.7), $\psi(ant, x, op_i)$ can be calculated as follows.

$$\psi(ant, x, op_i) = \begin{cases} U(p) & op_i \in p_{ant} \\ 0 & op_i \notin p_{ant} \vee p_{ant} \text{ violates Eq. 3.5-3.7} \end{cases} \tag{4.4}$$

The decay factor of the pheromone density due to evaporation is taken into consideration. Then other ants are allowed to choose their pathways till they reach the last leaf nodes one by one. The pheromone left on these pathways are updated. After the iterations finished, the optimal pathway, which has the maximum utility, can be now determined by choosing the pathway with maximum pheromone level.

The developed ACO-based algorithm is used to search an optimal pathway on the PDM that every ant has to go from the root node to the final nodes taking into consideration the cost, availability and quality of each operation, and then accumulate the cost, aggregate the availability and the quality while the ant is traveling to the final nodes. The developed ACO-based Algorithm can briefly be described in the procedure "ACOAlgorithm" of Algorithm 2, i.e., it computes the utility of the pathway traveled by each ant, and allocate a quantity amount of pheromone to the pathway, according to the utility of its pathway. Finally, the optimal pathway is retrieved which has the maximum utility.

Chapter 5

Evaluation of ACO methodology

To test the developed ACO algorithm in our research, we use the case studies. One is the model as shown in section 3.3, the other real life complex case study is described in section 5.2. The evaluation result shown in section 5.1 proves that the tool we developed for the research is executable and able to find optimal pathway in small size model. After that, we can see the real life business network is much complex than the previous examples in section 5.2. And then we will provide the parameters of reallife case study and test our developed ACO algorithm both on accuracy and time cost in section 5.3.

5.1 Ant Colony Optimization On Simple Case Study

In this section we are going to discuss the optimization result evaluation of developed ACO algorithm on the example provided in section 3.3. Three different requirements are tested based on 40 iterations and 100 ants in each iteration experiment test (should be big enough for the evaluation of the algorithm).

A-path is the availability optimal pathway of the PDM model without any side constraints; C_q-path is the cost optimal pathway with a minimum threshold of the quality level; $Q_{c,a}$-path is the quality optimal pathway with both a maximum constraint on cost and a minimum constraint on availability.

A-path: It is the optimization result when there is no side constraints, $w_q = 0$,

$w_a = 1$ and $w_c = 0$ in Eq. (3.3). In previous case study, the pathway with the highest availability, i.e. the highest probability of delivering the required end evaluation in the form of monitoring information (MON), is pathway 3. The total availability of this plan is 0.648. The optimization result is as follows in Figure 5.1.

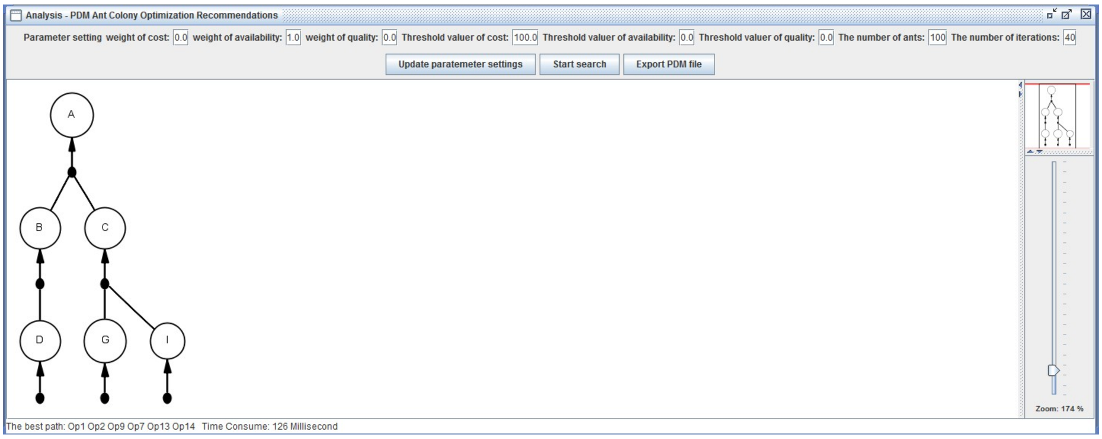

Figure 5.1: A-path optimization result in Ant colony algorithm.

C_q-path: It is the optimization result when $V_q(p) \geq \varphi_q$, $w_q = 0$, $w_a = 0$ and $w_c = 1$ in Eq. (3.3). In previous case study, a cost optimal pathway given a minimum quality level is tested. Suppose a consumer sets the threshold for the quality level to 0.7. Then, pathways 3, 4, 7, 8, 11, and 12 are to be considered. The pathway with the lowest cost is selected from this subset. So the cost optimal pathway given a minimum data quality level is pathway 7, with quality 0.7, and cost 2.2. The optimization result based on the developed ACO algorithm is as follows in Figure 5.2.

$Q_{c,a}$-path: The third scenario to be tested on case study model concerns the determination of the quality optimal pathway given maximum costs and a minimum level of availability.It is the optimization result when $V_a(p) \geq \varphi_a$, $V_c(p) \leq \varphi_c$, $w_q = 1$, $w_a = 0$ and $w_c = 0$ in Eq. (3.3). If the consumer has a budget constraint of at most 3.0 and wants to be for at least 35% sure that the monitoring infor-

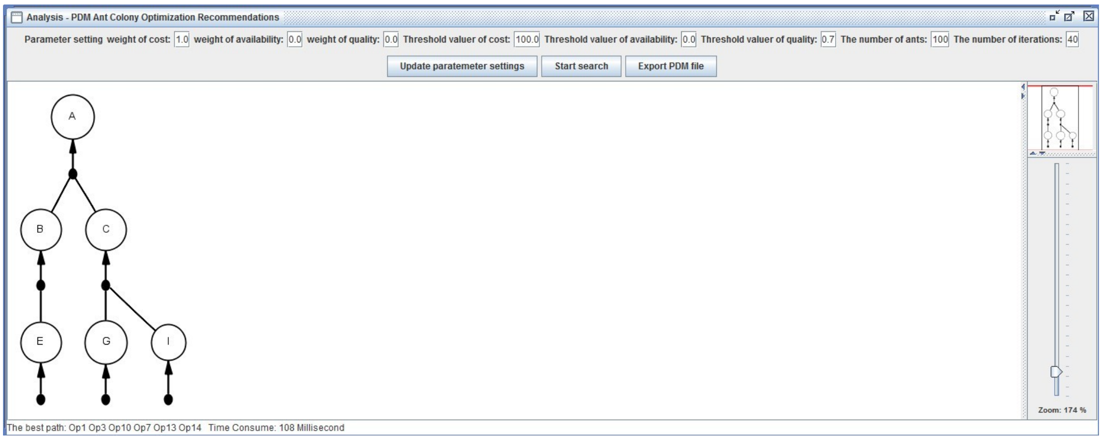

Figure 5.2: C_q-path optimization result in Ant colony algorithm.

mation is delivered, then the highest quality possible is achieved by pathway 9. The optimization result is as follows in Figure 5.3.

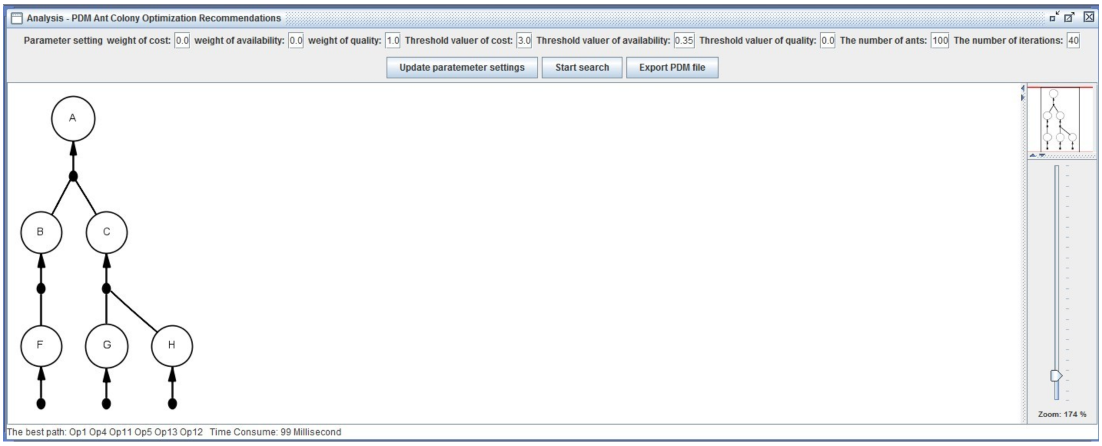

Figure 5.3: $Q_{c,a}$-path optimization result in Ant colony algorithm.

5.2 Real Life Complex Case Study Configuration

The second case study describes the monitoring process of awarding unemployment benefits in the Netherlands [30]. When a person becomes unemployed, he may be entitled to an unemployment benefit, which is part of the social security system in the Netherlands. An agency (the Dutch UWV) deals with the assessment of persons that have become unemployed and checks whether these persons meet all requirements to receive an unemployment benefit. The main regulations with regard to the decision to award unemployment benefits are laid down in the Dutch Unemployment Law. The agency also maintains operational interpretations of this law in handbooks. Furthermore, a detailed administration is kept of reasons for denying unemployment benefits to individual cases, as well as other statistical figures on the operations of the agency. Typical factors that are taken into account in the decision on awarding a unemployment benefit to an applicant are the reason for this person to have become unemployed, the length of the period that the previous job was held, and the coverage regulations. Figure 5.4 shows the PDM of this social benefits process and Table 5.1 contains a description of the data elements. The PDM contains 45 data elements and 50 operations. The details on the operation attributes can be found in Table 5.2.The unemployment benefits example is derived from an actual workflow process redesign case in industry.

5.3 Parameter Configuration and Algorithm Evaluation

Based on the model and data described above, from the top down consideration, we can conclude there are 12 complete pathways and their corresponding parameters as shown in Table 5.3. In this section we will prove ant colony optimization can work on this real life complex model with several alternatives and achieve the optimal pathway in a reasonable time.

The experiment consists of N replications, $N = 100$ in our case. The exper-

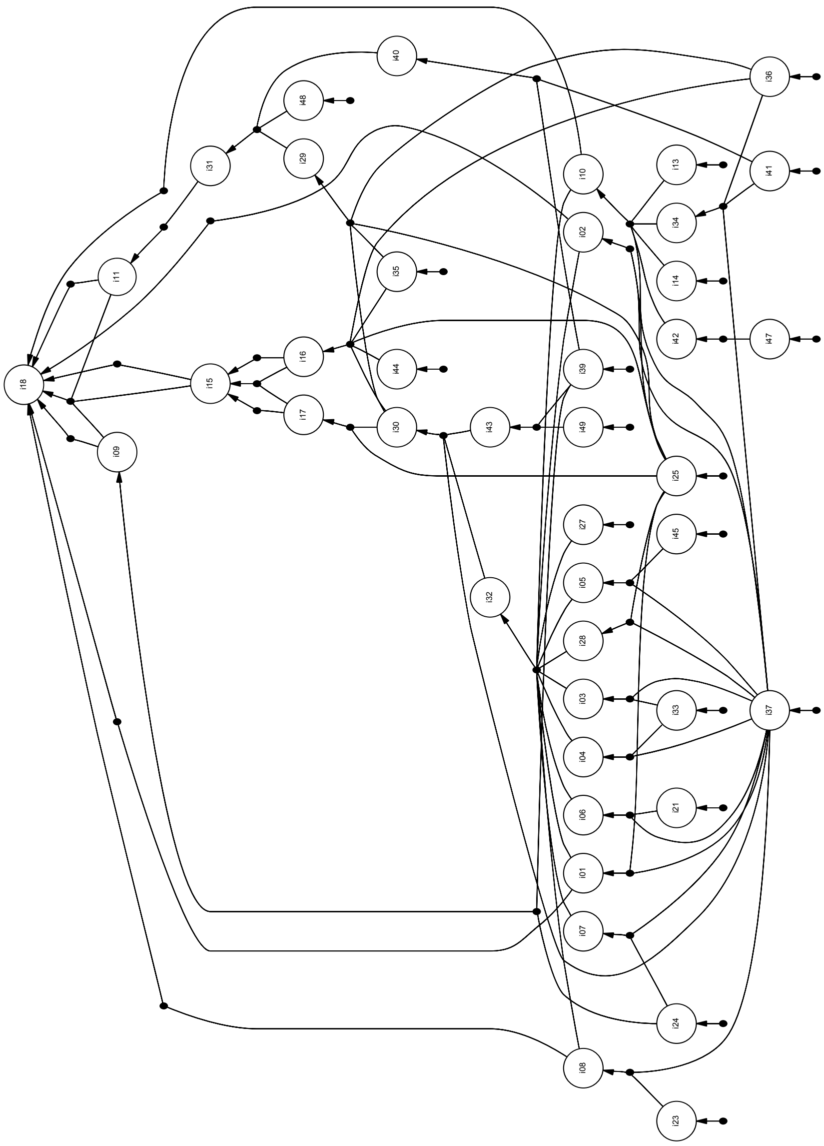

Figure 5.4: The PDM of the social benefits process

Table 5.1: Meaning of data elements of the PDM of the unemployment benefits example

Date element	information product statues
i01	Period in which claimant receives illness benefits
i02	Period in which claimant receives combined social benefits
i03	Period claimant lives/resides outside the Netherlands
i04	Period in which claimant does not rightfully live in the Netherlands
i05	Period in which claimant is detained/imprisoned
i06	Period in which claimant is 65 years or older
i07	Period in which claimant has legal scruples against insurance
i08	Period in which claimant enjoys holiday
i09	Period in which claimant is an employee
i10	Period in which claimant is unemployed
i11	Claimant satisfies refer requirement
i13	Date from which claimant lost the right for payment
i14	Date from which claimant is available to accept labor
i15	Claimant satisfies labor history requirement
i16	Claimants satisfies 4-out-of-5-years requirement
i17	Claim is directly following labor disablement benefits
i18	Claimant is entitled to (pay-related) unemployment benefits
i21	Birth date of claimant
i23	Claimants holiday administration
i24	Registration of unemployment insurance
i25	Registration of social benefits
i27	Claimants unemployment is caused by strike/work stoppage
i28	Period in which claimant receives re-integration benefits
i29	Refer period for claimant
i30	First day of unemployment of claimant
i31	Number of weeks claimant worked in refer period
i32	First week of unemployment of claimant
i33	Registration of housing
i34	Average number of labor hours per week of claimant
i35	First day of labor history for claimant
i36	Day status survey of claimants labor history
i37	Loss pattern of labor hours of claimant
i38	Care data on claimant
i39	Employment function of which the claimant has become unemployed
i40	Employment functions that have been followed up by the employment function of which the claimant has become unemployed
i41	Earlier employment function of the claimant
i42	Approved labor courses for unemployed
i43	Common first labor day for claimant
i44	List of claimants annual worked days
i45	Register of convictions
i47	Claimants courses that precede or follow on the loss of labor hours
i48	Weeks in refer period already taken into account
i49	Labor pattern of claimant
i50	Register of special classes of employment functions
i51	Claimant has taken care of under-age children

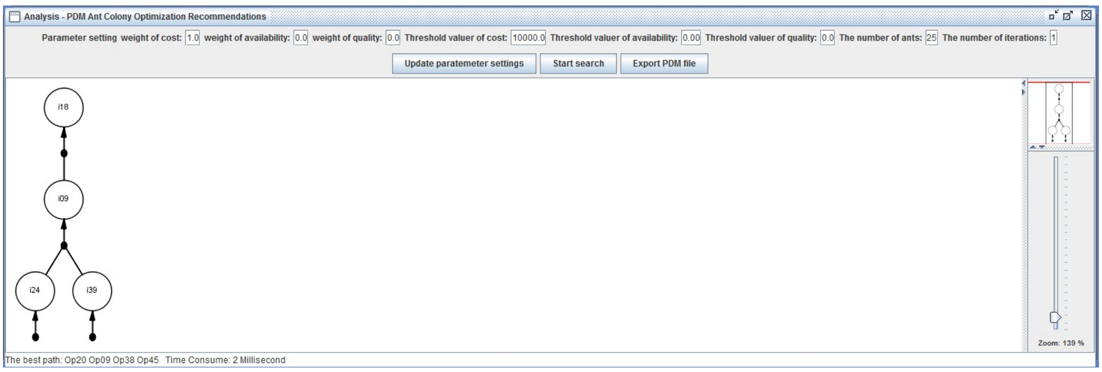

Figure 5.5: Minimum settings for cost optimization with ACO on case study

Table 5.2: Operation attributes for the unemployment benefits example

OperationID	Input	Output	Cost	Availability	Quality
Op01	i25 i27	i01	0	1	1
Op02	i25 i37	i02	0	1	0.8
Op03	i33 i37	i03	0	1	1
Op04	i33 i37	i04	0	1	1
Op05	i37 i45	i05	0	1	0.78
Op06	i21 i37	i06	0	1	1
Op07	i24 i37	i07	0	1	1
Op08	i23 i37	i08	0	1	0.6
Op09	i24 i39	i09	0	1	1
Op10	i13 i14 i34 i37 i42	i10	0	1	1
Op11	i31	i11	60	1	1
Op12	i16	i15	0	0.997	1
Op13	i17	i15	0	0.003	1
Op14	i16 i17	i15	0	1	1
Op15	i25 i30 i35 i36 i44	i16	561	1	1
Op16	i30	i17	0	1	0.88
Op17	i01	i18	0	0.009	0.68
Op18	i02	i18	0	0.013	1
Op19	i08	i18	0	0.016	1
Op20	i09	i18	0	0.002	1
Op21	i10	i18	0	0.068	1
Op22	i11	i18	0	0.019	0.95
Op23	i15	i18	0	0.210	1
Op24	i09 i11 i15	i18	0	1	1
Op25	i25 i37	i28	0	1	0.96
Op26	i25 i30 i35 i36	i29	0	1	1
Op27	i32 i37 i43	i30	0	1	0.84
Op28	i29 i40 i48	i31	0	1	1
Op29	i01 i02 i03 i04 i05 i06 i07 i08 i10 i27 i28	i32	0	1	1
Op30	i36 i37 i41	i34	420	1	0.79
Op31	i39 i41	i40	30	1	1
Op32	i47	i42	30	1	1
Op33	i39 i49	i43	60	1	1
Op34		i13	60	1	1
Op35		i14	8	0.8	1
Op36		i21	0	0.9	1
Op37		i23	67	0.6	1
Op38		i24	0	0.7	1
Op39		i25	0	0.5	1
Op40		i27	8	0.86	1
Op41		i33	0	0.79	1
Op42		i35	0	0.84	1
Op43		i36	100	0.91	1
Op44		i37	167	0.89	1
Op45		i39	17	0.65	1
Op46		i41	0	0.87	1
Op47		i44	0	0.78	1
Op48		i45	0	0.88	1
Op49		i47	33	0.83	1
Op50		i48	0	0.79	1
Op51		i49	0	0.95	1

Table 5.3: Complete pathways in the unemployment benefits example

PID	Operations	Data elements	Total Cost	Total Availability	Total Quality
1	Op08 Op19 Op37 Op44	i08 i18 i23 i37	234	0.008544	0.60
2	Op01 Op17 Op39 Op44	i01 i18 i25 i37	167	0.004005	0.68
3	Op09 Op20 Op38 Op45	i09 i18 i24 i39	17	0.000910	1.00
4	Op02 Op18 Op39 Op44	i02 i18 i25 i37	167	0.005785	0.80
5	Op21 Op10 Op35 Op32 Op49 Op34 Op30 Op46 Op44 Op43 Op44	i18 i10 i42 i47 i13 i14 i34 i38 i41 i37	985	0.028315	0.79
6	Op22 Op11 Op28 Op26 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op42 Op39 Op43 Op50 Op31 Op45 Op46	i18 i11 i31 i29 i35 i30 i37 i32 i43 i01 i02 i03 i04 i05 i06 i07 i24 i08 i21 i33 i45 i10 i27 i28 i39 i49 i25 i36 i48 i40 i41	1636	0.000002	0.78
7	Op23 Op13 Op16 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45	i18 i15 i17 i25 i30 i37 i32 i43 i01 i02 i03 i04 i05 i06 i07 i24 i08 i21 i33 i45 i10 i27 i28 i39 i49	1429	0.000014	0.78
8	Op23 Op12 Op15 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op47 Op42 Op43	i18 i15 i16 i30 i37 i32 i43 i01 i02 i03 i04 i05 i06 i07 i24 i08 i21 i33 i45 i10 i27 i28 i39 i49 i35 i36 i44	2090	0.002774	0.78
9	Op23 Op14 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op47 Op42 Op43	i18 i15 i16 i17 i30 i37 i32 i43 i01 i02 i03 i04 i05 i06 i07 i24 i08 i21 i33 i45 i10 i27 i28 i39 i49 i35 i36 i44	3519	0.000013	0.78
10	Op24 Op11 Op28 Op26 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op42 Op39 Op43 Op50 Op31 Op45 Op46 Op09 Op38 Op45 Op13 Op16 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45	i18 i09 i24 i39 i11 i31 i29 i35 i30 i37 i32 i43 i01 i02 i03 i04 i05 i06 i07 i24 i08 i21 i33 i45 i10 i27 i28 i39 i49 i25 i36 i48 i40 i41 i15 i17 i25 i30 i37 i32 i43	3082	0.000000	0.78
11	Op24 Op11 Op28 Op26 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op42 Op39 Op43 Op50 Op31 Op45 Op46 Op09 Op38 Op45 Op12 Op15 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op47 Op42 Op43	i18 i09 i24 i39 i11 i31 i29 i35 i30 i37 i32 i43 i01 i02 i03 i04 i05 i06 i07 i08 i21 i33 i45 i10 i27 i28 i49 i25 i36 i48 i40 i41 i15 i16 i17 i44	3682	0.000000	0.78
12	Op24 Op11 Op28 Op26 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op42 Op39 Op43 Op50 Op31 Op45 Op46 Op09 Op38 Op45 Op14 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op27 Op44 Op29 Op33 Op07 Op38 Op44 Op01 Op39 Op44 Op06 Op36 Op44 Op04 Op41 Op44 Op05 Op48 Op44 Op03 Op41 Op44 Op25 Op39 Op44 Op40 Op51 Op45 Op47 Op42 Op43	i18 i09 i24 i39 i11 i31 i29 i35 i30 i37 i32 i43 i01 i02 i03 i04 i05 i06 i07 i08 i21 i33 i45 i44 i10 i27 i28 i39 i49 i25 i36 i48 i40 i41 i15 i37 i16 i17 i30	5082	0.000000	0.78

iment is evaluated using two parameters, i.e. accuracy (acc) and quality ($qual$). We also evaluated the confidence interval of quality. The explanation of the evaluation parameter is as follows.

$$acc_n = \begin{cases} 1 & \text{output of this replecation is the optimal pathway} \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

$$\text{Accuracy} = \frac{\sum_{n=1}^{N} acc_n}{N} * 100\% \tag{5.2}$$

Where acc_n is the accuracy for each test of the optimization tool and *Accuracy* is the average accuracy of the experiment. The Accuracy can only test if for each test the parameter settings can get the optimal pathway, but not the distance from average utility to the utility of optimal pathway. Therefore, we define the following quality of experiment to express how far is the result from optimal pathway in utility.

$$\text{Quality} = \frac{\sum_{n=1}^{N} U_n/U_b}{N} * 100\% \tag{5.3}$$

Where U_b is the utility of the optimal pathway in this requirement from stakeholder, U_n is the utility of the optimization result for each test. The accuracy and quality are both test result about average standard of level the method can get to the optimal pathway based on different settings. A $(1-\alpha)\%$ confidence interval for the mean is a range of quality values running from a lower bound (LB) to an upper bound (UB) for which we can be $(1-\alpha)\%$ confident that the true population mean falls.

$$\text{Confidence Interval} = \overline{Q} \pm (t_{df,1-\alpha/2})\frac{s}{\sqrt{N}} \tag{5.4}$$

$$\text{where s = the standard deviation:} \sqrt{\frac{\sum_{n=1}^{N}[Q_n - \overline{Q}]^2}{N-1}} \tag{5.5}$$

Where Q_n is the quality of number n optimization test which means U_n/U_b, $(t_{df,1-\alpha/2})$ is the standard number for confidence interval extracted from student's t distribution. In our experiment, we choose $(1-\alpha)\% = 95\%$ under $N = 100$ times test, therefore $(t_{df,1-\alpha/2}) = 2.01$.

Table 5.4: Parameters in ACO ants configuration

Iteration	Ants number	Accuracy	Quality	Confidence interval	Time consuming per test
40	100	100%	100%	100% ± 0%	534millisecond
1	25	100%	100%	100% ± 0%	2millisecond
1	20	98%	99.61%	99.61% ± 0.45%	2millisecond
1	10	80%	98.33%	98.33% ± 0.85%	≪1millisecond
1	5	20%	91.57%	91.57% ± 4.08%	≪1millisecond

The experiment is set to test the different parameters for 100 times ($N = 100$) on optimization the lowest cost pathway ($w_c = 1$, $w_a = 0$, $w_q = 0$ in Eq. (3.3) in the real life case study. It is easily found out from Table 5.3 that the lowest cost is 17 from pathway 3 compose with $\{op09, op20, op38, op45\}$.Based on 100 times test, the minimum parameter that works on this model is 25 ants for 1 iteration or 5 ants for 5 iterations as shown in Figure 5.5. The time cost for these two optimization is 2 milliseconds and 9 milliseconds. For lower number of settings for ants or iterations, the experiment result is not guaranteed to get the optimal pathway. In the test, we can find out that with 20 or 10 ants in 1 iteration, or with 5 ants in 4 or 3 iterations, the optimized result can be mistakenly calculated as pathway 2 or pathway 4 (cost=167). With only 5 ants in 2 iteration, the optimized result sometimes is pathway 1 (cost=234).With only 5 ants in 1 iteration, the optimized result sometimes is even pathway 5 (cost=985). This implies that lower number of ants or iterations means less possibility of covering the real optimal pathway in the PDM model. The result accuracy, quality, confidential interval and time consuming comparation are considered in our experiment result.The camparation with a stable iteration number (Iteration=1) and different ants number setting is shown in Table 5.4. On the other hand, if we set a stable ants number (Ants number=5) and decrease the iteration number setting, the result is shown in Table 5.5. The detail experiment result is provided in Appendix A. The accuracy experiment result comparation is as shown in Figure 5.6. The quality experiment result comparation with confidence interval is as shown in Figure 5.7.

Another parameter should be considered is β parameter in pheromone calculation. Based on the experiment, when ants and iterations setting is big enough to cover the model size, β does not influence the experiment result generally. This is because based on our case study model size, 12 pathways can be always covered

Table 5.5: Parameters in ACO iterations configuration

Iteration	Ants number	Accuracy	Quality	Confidence interval	Time consuming per test
40	100	100%	100%	100% ± 0%	534millisecond
5	5	100%	100%	100% ± 0%	9millisecond
4	5	97%	99.50%	99.50% ± 0.55%	7millisecond
3	5	95%	99.17%	99.17% ± 0.68%	4millisecond
2	5	78%	94.10%	94.10% ± 0.44%	3millisecond
1	5	20%	91.57%	91.57% ± 4.08%	≪1millisecond

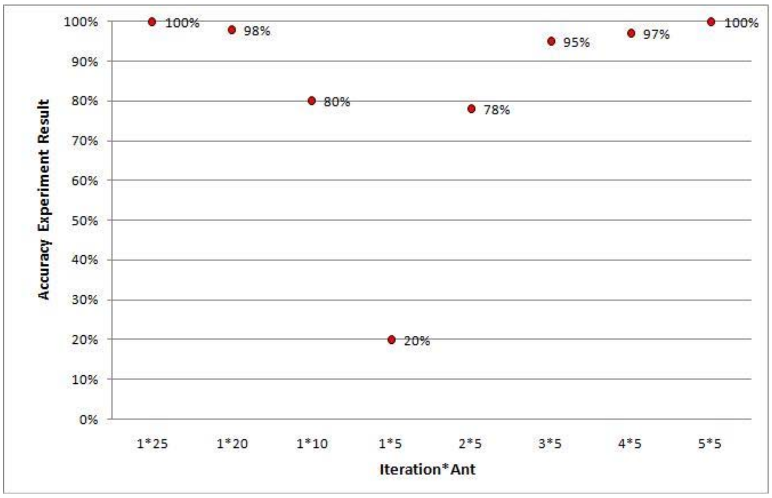

Figure 5.6: Accuracy experiment result in ACO configuration experiment result

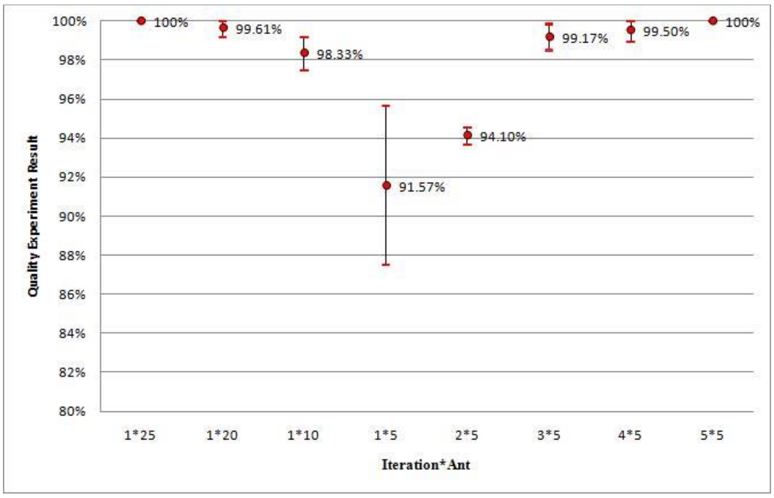

Figure 5.7: Quality experiment result in ACO configuration experiment result

in 1 iteration with 25 ants or 5 iterations with 5 ants setting. But when the ants and iterations number are low, such as 5 ants with 1 iteration in our case study, the experiment result is different. When $\beta = 0.1$ or $\beta = 1.0$, the result accuracy is generally around the result when $\beta = 0.7$. But the wrong optimization results includes more low utility pathway such as pathway 5 (cost=985) which reduce the quality of the optimization result when pathways can not be all covered based on the PDM model size.

Chapter 6

Implementing Optimization and Execution

In this chapter we discuss the implementation of the methodology proposed in this paper. In section 6.1, the introduction of the general business process monitoring framework and emphasis on the importance of the research in this thesis are provided. The general introduction of ProM, the implementation platform of our research is provided in section 6.2. After that, we introduce the plug-ins about PDM and especially the plug-ins we implemented for optimization the pathway of PDM in section 6.2. Finally the execution of the ACO optimization result pathway is discussed with execution example in section 6.4.

6.1 Business Process Monitoring Framework

The framework depicted in Figure 6.1 shows the how the research of the thesis can contribute in the improvement of business network process monitoring. We can see that implementation in ProM framework can support the creation of an monitoring PDM model, optimization of the best pathway in the model, and execution of the optimization result based on the requirement of the monitoring stakeholder. As depicted in the framework, each operation in the PDM is implemented by a Web service published in the Service Registry. The Monitoring Enactment component receives the complete pathway from the MON-PDM Op-

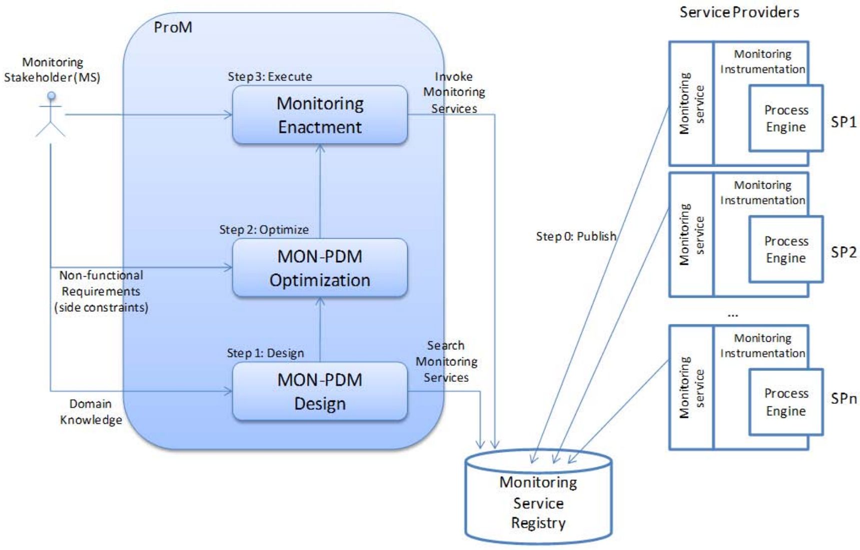

Figure 6.1: The Business Process Monitoring Framework

timization component. Each operation in the PDM contains a reference to the Web service implementing it.

The research scope of this thesis if restricted on step 2 and 3 in Figure 6.1. What we realize is the ACO optimization of the pathway in the PDM model and execution of the optimal pathway PDM result inside the ProM platform. These two steps are quite important in the general research framework as optimization is the basement of the monitoring suggestion and execution implementation proves the executable property of the research.

6.2 PROM

ProM is the most promising process mining framework available, which supports the implementation of process mining techniques [1, 2, 34]. It is an extensible framework that supports a wide variety of process mining techniques in the form of plug-ins [29]. ProM is widely used in many process mining research area. The architecture of ProM is set up in such a way that it is easy to include newly developed algorithms and implementations without modifying and recompiling the ProM framework. Therefore, it is a truly pluggable environment. Each functional part of implementation is called a plug-in in ProM terms. Currently, there are already more than 250 plug-ins available, supporting for instance, the import of several processing modeling languages, such as: Petri Net, YAWL (Yet Another Workflow Language), BPEL (Business Process Execution Language), EPML (Event-driven Process Chains Model Language) [30]. The modeling language we use in our research is PDM (Product Data Model) which is also supported by ProM.

Also, ProM includes different plug-ins for different functions on process. The mining plug-ins includes: control-flow mining techniques plug-ins (such as the Alpha algorithm, Genetic mining, Multi-phase mining) [27, 33]; analysis plug-ins of the organizational perspective (such as the Social Network miner, the Staff Assignment miner) [28]; Plug-ins dealing with the data perspective (such as the Decision miner) [24]. Furthermore, there are analysis plug-ins dealing with: verification of process models (e.g., Woflan analysis) [32]; verification of Linear Temporal Logic (LTL) formulas on a log; checking the conformance between a given process model and a log [26]; performance analysis (Basic statistical analysis, and Performance Analysis with a given process model) [30].

6.3 PDM related plug-ins

As mentioned in the previous section, PDM is one of the modeling language that is supported by ProM. There is one import plug-in especially for ProM to read the PDM model file. After importing the PDM file, there exists several conversion and analysis plug-ins for PDM. There are seven algorithms based on different constraints which have been presented in [30] as conversion plug-ins in ProM. Each plug-in takes a PDM as input and produces the process model (represented by a Petri net or by a YAWL model) as output.

Also ,the ProM framework includes several analysis plug-ins of PDM. For example, “PDM recommendations ” plug-in is for supporting the direct execution of a PDM by guiding the user through all steps to the end product by the simple strategies expressed in [30]. Furthermore, the Markov decision strategies are implemented in the recommendation tool as plug-in “PDM MDP Statespace”. There are other plug-ins for analysis of PDM. But all these tools can not provide pathway optimization based on predefined monitoring parameters as the requirement discussed is section 2.1. Therefore, the implementation of our research is needed.

Our research implementation is about optimization pathway in PDM model for cross-organization monitoring and the execution of the optimization result, which belongs to PDM analysis. As shown in Figure 6.2, we implemented two plug-ins in the analysis section of PDM. The first one is "PDM Simple Recommendations", which is based on the local optimization algorithm discussed in section 3.2. The other plug-in is "PDM Ant Colony Optimization Recommendation" based on the ant colony optimization algorithm discussed in section 3.3.

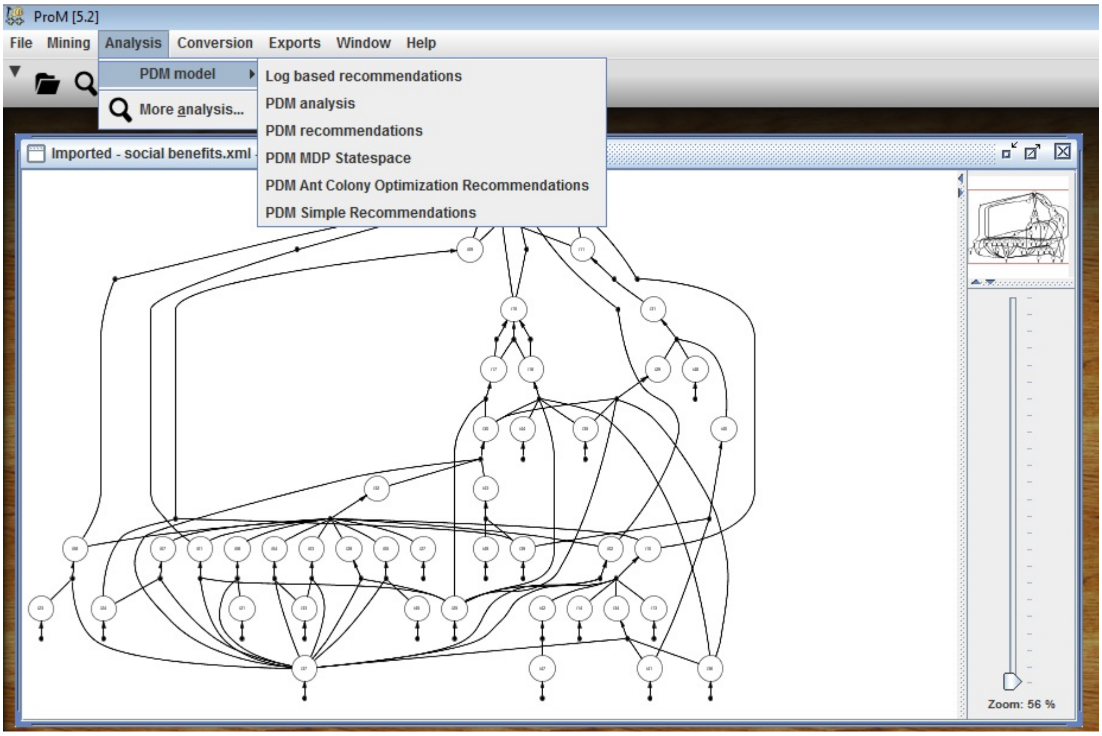

Figure 6.2: Plug-in implementation in ProM

6.4 Execution of optimization result

After optimization of the pathways in the PDM model, the next step in our research is executing the result as shown in the research procedures in Figure 6.1. Based on the optimal pathway, we should have a web service to execute from the leaf data elements, along the whole complete pathway, to the root data element of the product data for the business model. In this way, we can make the elementary monitoring information products that can be e.g. the progress information available by actors in the business network to other actors for monitoring purposes[9]. This is realized in the plug-in of ProM as a form of web service with awareness of the step in the optimized pathway as shown in Figure 6.3. When considering the evaluation example, the optimal cost pathway is $\{op09, op20, op38, op45\}$. Based on the data elements explanation of Table 4, it means the leaf data elements are gathering the employment function of which the claimant has become unemployed (with cost of 17) and the registration of unemployment insurance. After that, the period in which claimant is an employee is considered. The result of these three data elements leads to the product that claimant is entitled to (pay-related) unemployment benefits. The status shown in Figure. 20 express the execution steps of the optimization result with both the interface and the visualization of temporary status in the PDM pathway. The figures of ACO implementation interface and execution of the optimization result in ProM are provided in Appendix B.

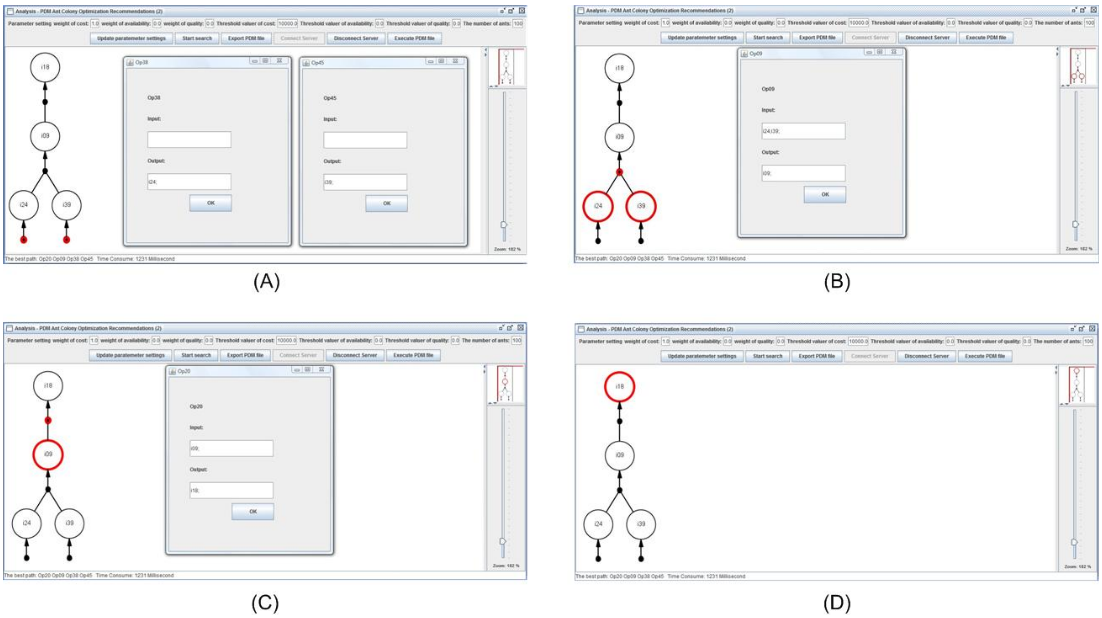

Figure 6.3: Execution step of result pathway

Chapter 7

Conclusions and Future Work

In this thesis we present an innovative application of PDM optimization methodology, that is, the developed ant colony optimization of monitoring processes in collaborative business networks. The innovation brought about by this thesis is twofold. On the one hand, to our best knowledge, the algorithm of optimization the optimal pathway in PDM model is not defined in previous researches. In our research, we adapt ant colony optimization (ACO) to find the optimal pathway in the PDM model. On the other hand, we realize the implementation of both the optimization in ProM 5.2 and the execution of the optimization result based on online interaction with the client. These implementation realize the evaluation of the methodology and proof of the execution possibility.

There are several remaining challenges that need further research. Firstly, it would be more convincing to test our methodology with more complex PDM models. So in the future we need more data on the PDM model of monitoring process in business network to verify our proposal. Also, future work will concern the extension and refinement of the product data model parameter settings. In particular, we plan to consider additional non-functional dimensions, and more complex utility functions such as different parameters influencing each other instead of existing independently. Moreover, the ACO algorithm is a heuristic algorithm. When the PDM model size is much bigger than the provided case studies, more computation time is needed to guarantee the optimal result in our algorithm. Configuration of the optimization ability for ant and iteration settings may help in applying our methodology on business process optimization.

Appendix A

The experiment result of Section 5.3 are as follows, the corresponding data is provided in table 5.3:

Table A.1: Experiment result of developed ACO algorithm

Iteration*Ants	Pathway	Times	Cost
40*100	pathway 3	100	17
1*25	pathway 3	100	17
1*20	pathway 3	98	17
	pathway 2/4	2	167
1*10	pathway 3	80	17
	pathway 2/4	20	167
1*5	pathway 3	20	17
	pathway 2/4	56	167
	pathway 1	13	234
	pathway 5	11	985
2*5	pathway 3	80	17
	pathway 2/4	20	167
	pathway 1	2	234
3*5	pathway 3	95	17
	pathway 2/4	5	167
4*5	pathway 3	97	17
	pathway 2/4	3	167
5*5	pathway 3	100	17

Appendix B

The Screenshots of the implementation interfaces are as follows:

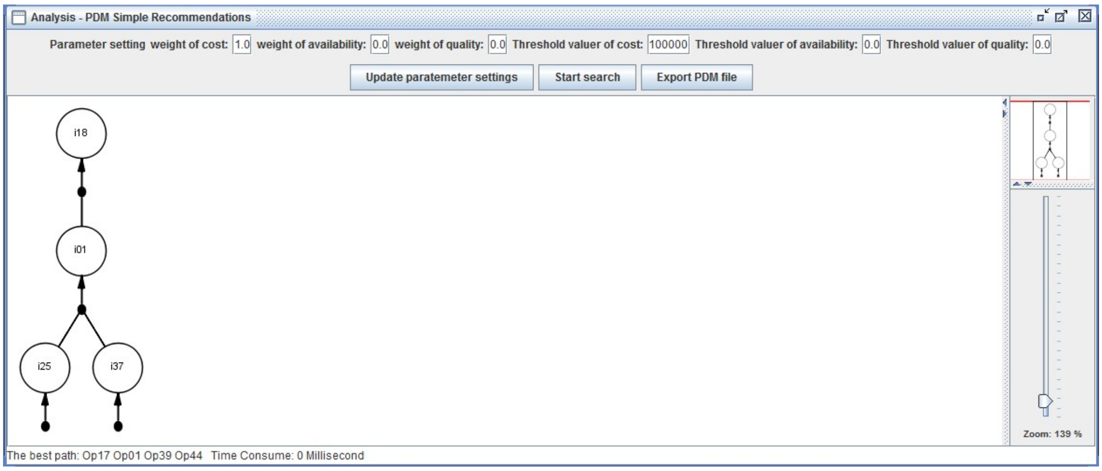

Figure B.1: Simple local greedy optimization interface

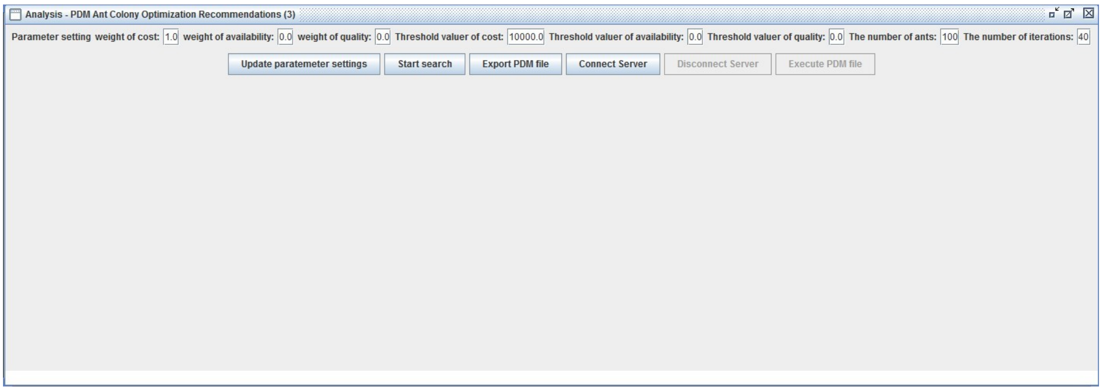

Figure B.2: ACO optimization interface

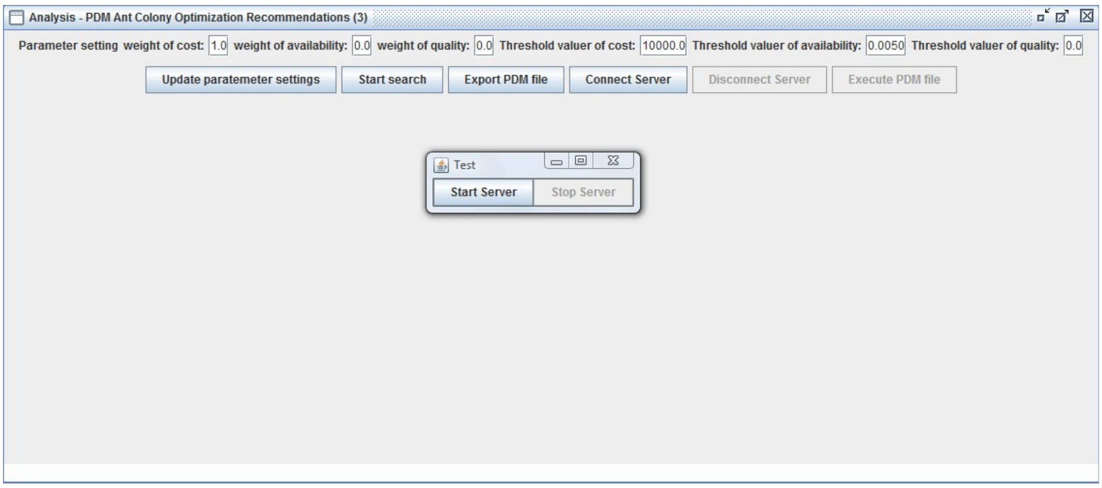

Figure B.3: Start client server to execute the ACO optimization result

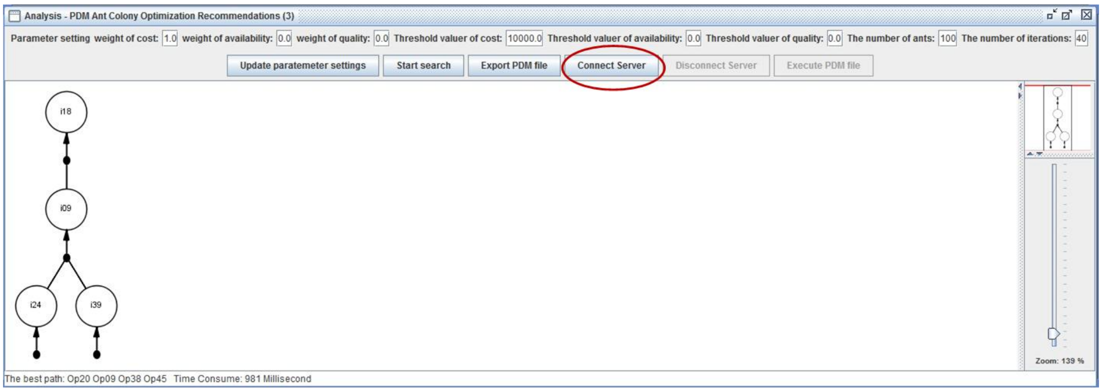

Figure B.4: Connect ProM with the server for collaboration

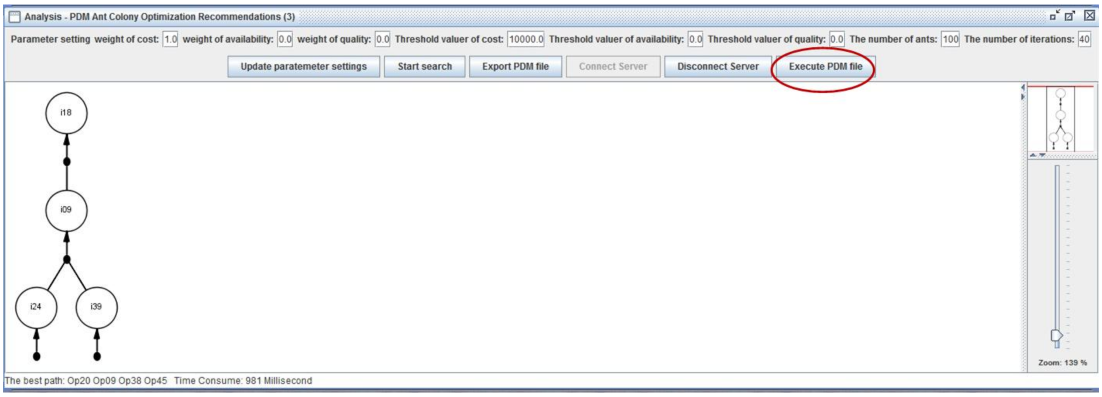

Figure B.5: Press the execute PDM result button to execute the optimization result

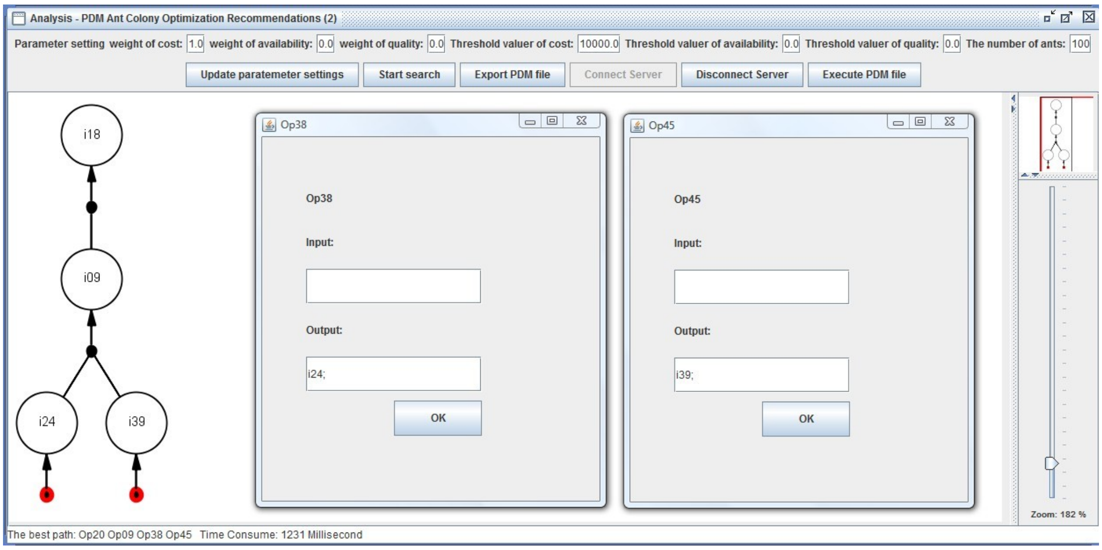

Figure B.6: Two operations available without input data element

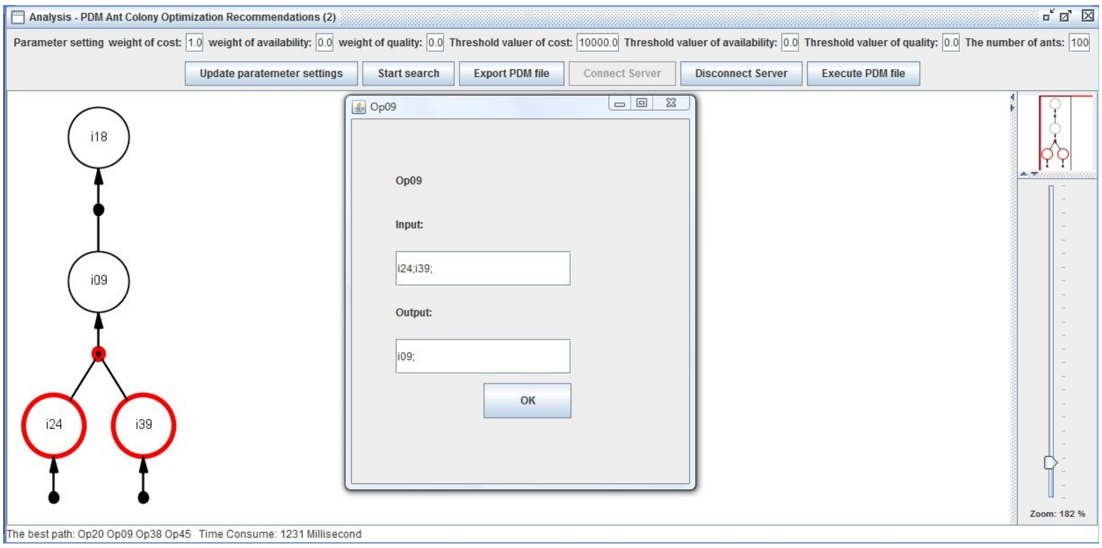

Figure B.7: One operation available with two input data elements

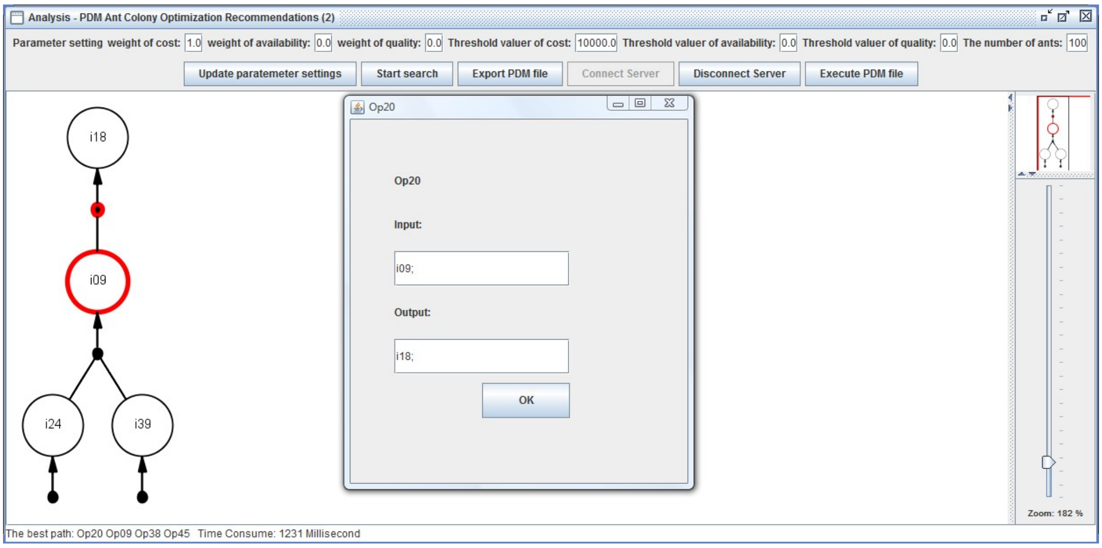

Figure B.8: One operation available with one input data element

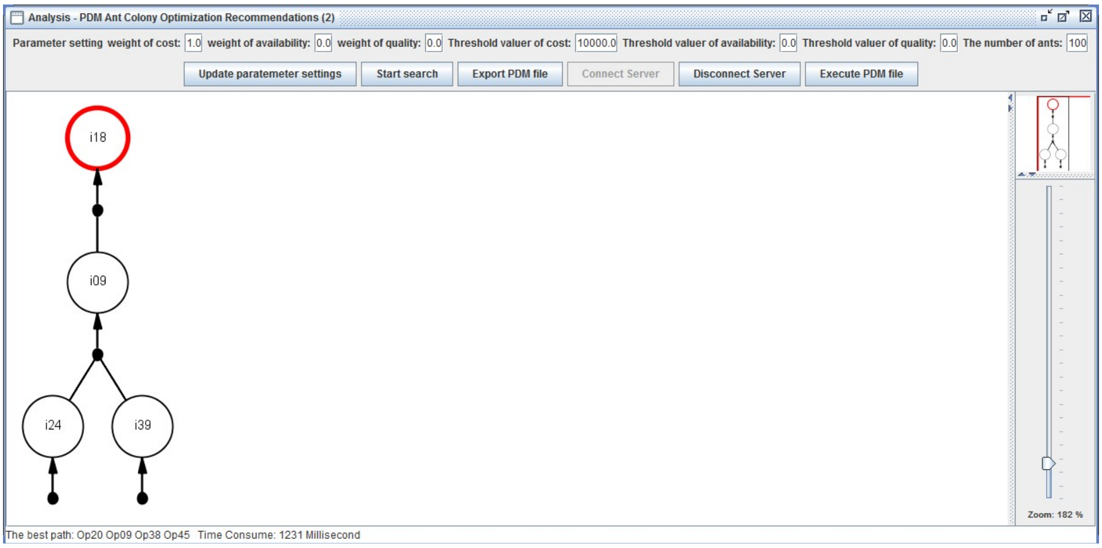

Figure B.9: End information product available

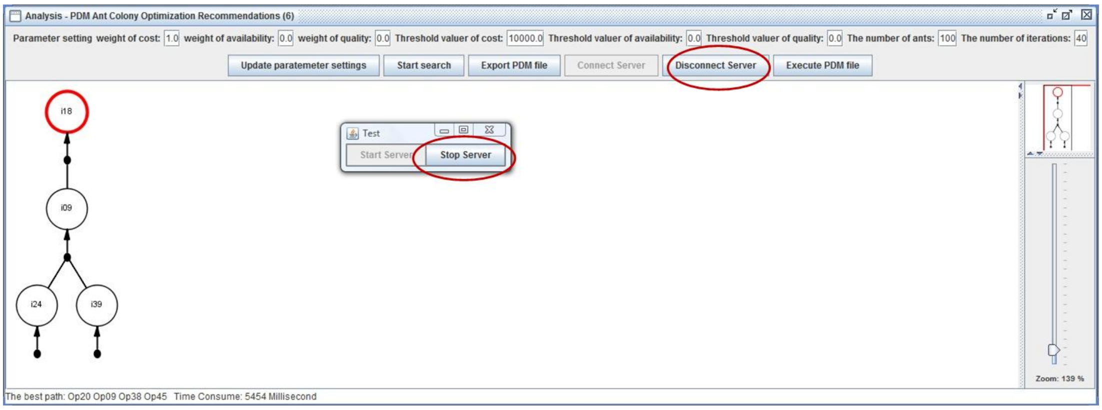

Figure B.10: Stop the connection between ProM and client server

References

[1] *Process Mining Home Page*. http://www.processmining.org. 49

[2] *ProM Framework*. http://prom.sourceforge.net. 49

[3] Michael Alles, Gerard Brennan, Alexander Kogan, and Miklos A. Vasarhelyi. Continuous monitoring of business process controls: A pilot implementation of a continuous auditing system at siemens. *International Journal of Accounting Information Systems*, 7(2):137 – 161, 2006. v, 5, 7

[4] Danilo Ardagna and Barbara Pernici. Adaptive service composition in flexible processes. *IEEE Transactions on Software Engineering*, 33:369–384, 2007. 21

[5] Benoit A. Aubert, Michel Patry, and Suzanne Rivard. A framework for information technology outsourcing risk management. *SIGMIS Database*, 36:9–28, October 2005. 4

[6] Ugo Chirico. *A Java Framework for Ant Colony Systems*. http://www.ugosweb.com/Documents/jacs.aspx. 31, 32

[7] Dickson K.W. Chiu, Kamalakar Karlapalem, Qing Li, and Eleanna Kafeza. Workflow view based e-contracts in a cross-organizational e-services environment. *Distributed and Parallel Databases*, 12:193–216, 2002. 10.1023/A:1016503218569. 3

[8] M. Comuzzi, J. Vonk, and P. Grefen. Continuous monitoring in evolving business networks. In *Proc. CoopIS*, pages 168–185, 2010. 1, 19

[9] Marco Comuzzi and Irene Vanderfeesten. Product-based workflow design for monitoring of collaborative business processes. In *23rd International Conference on Advanced Information Systems Engineering*, page to appear, 2011. v, 1, 12, 13, 15, 52

[10] M. Dorigo, M. Birattari, and T. Stuzle. Ant colony optimiaztion. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006. 29, 30

[11] M. Dorigo and C. Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243 – 278, 2005. 29, 30

[12] M. Dorigo and T. Stuzle. *Ant Colony Optimization.* MIT Press, USA, 2004. 29, 30

[13] Paul Grefen, Karl Aberer, Yigal Hoffner, and Heiko Ludwig. CrossFlow: cross-organizational workflow management in dynamic virtual enterprises. *Computer Systems Science and Engineering*, 5:277–290, 2000. v, 4, 5, 8

[14] G. Gutin, A. Yeo, and A. Zverovich. Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics*, 117:81–86, 2002. 28

[15] Eric van Heck and Peter Vervest. Smart business networks: how the network wins. *Commun. ACM*, 50:28–37, June 2007. 3

[16] Patrick R. McMullen. An ant colony optimization approach to addressing a JIT sequencing problem with multiple objectives. *Artificial Intelligence in Engineering*, 15(3):309 – 317, 2001. 29

[17] Tova Milo and Daniel Deutch. Querying and monitoring distributed business processes. *Proc. VLDB Endow.*, 1:1512–1515, August 2008. 5

[18] R.J. Mullen, D. Monekosso, S. Barman, and P. Remagnino. A review of ant algorithms. *Expert Systems with Applications*, 36(6):9608 – 9617, 2009. 29

[19] R.F.T. Neto and M.G. Filho. A software model to prototype ant colony optimization algorithms. *Expert Systems with Applications*, 38(1):249 – 259, 2011. 29

[20] A Orlicky. Structuring the bill of materials for MRP. *Production and Inventory Management*, 13:19–42, December 1972. 9

[21] H. Panahi and R.T. Moghaddam. Solving a multi-objective open shop scheduling problem by a novel hybrid ant colony optimization. *Expert Systems with Applications*, 38(3):2817 – 2822, 2011. 29

[22] Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, and Frank Leymann. Service-oriented computing: A research roadmap, 2008. 4

[23] Hajo A. Reijers, Selma Liman, and Wil M.P. van der Aalst. Product-based workflow design. *Management Information Systems / Summer*, 20(1):229–262, 2003. v, 9, 11, 14

[24] A. Rozinat and W. van der Aalst. Decision mining in prom. In Schahram Dustdar, Jos Fiadeiro, and Amit Sheth, editors, *Business Process Management*, volume 4102 of *Lecture Notes in Computer Science*, pages 420–425. Springer Berlin, Heidelberg, 2006. 50

[25] Ajaya Kumar Tripathy and Manas Ranjan Patra. An event based, non-intrusive monitoring framework for web service based systems. In *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*, pages 547 – 552. v, 5, 6

[26] W. van der Aalst, H. de Beer, and B. van Dongen. Process mining and verification of properties: An approach based on temporal logic. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems 2005: CoopIS, DOA, and ODBASE*, volume 3760 of *Lecture Notes in Computer Science*, pages 130–147. Springer Berlin, Heidelberg, 2005. 50

[27] W. van der Aalst, A. de Medeiros, and A. Weijters. Genetic process mining. In *Applications and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 985–985. Springer Berlin, Heidelberg, 2005. 50

[28] Wil M. P. van der Aalst, Hajo A. Reijers, and Minseok Song. Discovering social networks from event logs. *Computer Supported Cooperative Work (CSCW)*, 14:549–593, 2005. 50

[29] B. van Dongen, A. de Medeiros, H. Verbeek, A. Weijters, and W. van der Aalst. The ProM framework: A new era in process mining tool support. In Gianfranco Ciardo and Philippe Darondeau, editors, *Applications and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 1105–1116. Springer Berlin / Heidelberg, 2005. 49

[30] Vanderfeesten and Irene T.P. *Product-Based Design and Support of Workflow Processes*. PhD thesis, Technology University of Eindhoven, Department of Mathematics & Computing Science, 2009. 39, 50

[31] Irene Vanderfeesten, Hajo A. Reijers, and Wil M.P. van der Aalst. Product-basedworkflow support. *Information Systems*, 36:517–535, 2011. 14

[32] H. M. W. Verbeek, T. Basten, and W. M. P. van der Aalst. Diagnosing workflow processes using woflan. *The Computer Journal*, 44(4):246–279, 2001. 50

[33] A. J. M. M. Weijters and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16:2004, 2004. 50

[34] A.J.M.M. Weijters, W.M.P. van der Aalst, B.F. van Dongen, R. Mans C. Gnther, A.K. Alves de Medeiros, A. Rozinat, M. Song, and H.M.W. Verbeek. Process mining with ProM. In *Proceedings of the 19th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC)*, 2007. 49