

# Making decision process knowledge explicit using the product data model

***Citation for published version (APA):***

Petrusel, R., Vanderfeesten, I. T. P., Dolean, C., & Mican, D. (2011). *Making decision process knowledge explicit using the product data model*. (BETA publicatie : working papers; Vol. 340). Technische Universiteit Eindhoven.

***Document status and date:***

Published: 01/01/2011

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

## **Making Decision Process Knowledge Explicit Using the Product Data Model**

Razvan Petrusel, Irene Vanderfeesten,  
Cristina Claudia Dolean, Daniel Mican

Beta Working Paper series 340

BETA publicatie	WP 340 (working paper)
ISBN	978-90-386-2469-3
ISSN	
NUR	982
Eindhoven	March 2011

# Making Decision Process Knowledge Explicit Using the Product Data Model

Razvan Petrusel<sup>1</sup>, Irene Vanderfeesten<sup>2</sup>, Cristina Claudia Dolean<sup>1</sup>, Daniel Mican<sup>1</sup>

<sup>1</sup> Faculty of Economical Sciences and Business Administration, Babes-Bolyai University,  
Teodor Mihali str. 58-60, 400591 Cluj-Napoca, Romania  
{razvan.petrusel, cristina.dolean, daniel.mican}@econ.ubbcluj.ro

<sup>2</sup> School of Industrial Engineering, Eindhoven University of Technology, PO Box 513, 5600  
MB Eindhoven, The Netherlands  
i.t.p.vanderfeesten@tue.nl

**Abstract.** In this paper, we present a new knowledge acquisition and formalization method: the decision mining approach. Basically, we aim to produce a model of the workflow of mental actions performed by decision makers during the decision process. We show that through the use of a Product Data Model (PDM) we can make explicit the knowledge employed in decision making. We use the PDM to provide insights into the data view of a business decision process. To support our claim we introduce our complete, functional decision mining approach. We present a “decision-aware system” that introduces the user in a simulation scenario environment containing all data needed for the decision. We log the interaction with the system (focusing on data manipulation and aggregation) and output a user action log file. The log file is then mined through the presented mining algorithm and a Product Data Model (PDM) is created. The advantage of our approach is that, when needed to investigate a large number of subjects, it is much faster, less expensive and produces more objective results than classical knowledge acquisition methods (such as interview and questionnaires). The feasibility and usability of our approach is shown by a prototype, a case study and experiments.

**Keywords:** Decision Mining, Product Data Model, Decision-aware System, Decision Workflow.

## 1. Introduction

In the area of financial decisions there are a lot of different approaches that generate fuzzy decision processes. Our experience shows that some managers disregard some data elements not just because they consider it important but because it just slipped their mind or they just don't know about it. For example, when managers intend to contract a loan, some of them may consider important the amount cashed from customers in the previous months while others may not or may disregard this data item completely due to various reasons. People may also perform decision making in unstructured situations by using feelings, intuition, etc.

The decision process and the choice regarding decision alternatives have been researched early in the 60's. The root of current well known decision processes in literature is Simon's model [Simon, 1960]. It is composed of four phases: (i)

intelligence gathering, (ii) design, (iii) choice, and (iv) implementation. This initial classification was later expanded by other researchers but Simon's basic process can be found at the core of them all. The focus of those approaches was on producing several decision alternatives and on how the choice of one alternative needs to be performed. Less attention was given to identifying relevant information for the decision at hand or how to manipulate all the data items that need to be considered when making the actual choice. All those approaches also assume that data needed for the decision process is available and the user knows which data items are needed to make a well informed decision [Turban, 2010]. We argue that a person making a decision does not always know which information is needed or relevant and does not have a clear overview of how available data should be aggregated.

We are aiming to provide a better insight into the decision process by making the implicit knowledge used in the decision process explicit. We are looking at different persons performing the same decision and we try to evaluate the process that they perform in order to make a decision. This involves a lot of mental activities which we need to capture and to make explicit in a model. Therefore, we need to find a graphical representation that can be presented and easily understood by persons with less domain knowledge. We propose to use a Product Data Model (PDM) as a graphical representation that can depict the data aggregation used in the decision process and that can be easily understood even by untrained decision makers.

The aim of this paper is to show how a model explicitly depicting the knowledge behind the data used in a decision making process is created. Our approach includes all the necessary steps to automatically mine such a model based on the interaction of the decision maker with software. The framework includes a 'decision-aware system', a mining algorithm and the PDM format for representing the mined knowledge.

The overall goal of this approach is to enable an untrained decision maker to follow a decision process extracted from expert users in a specific domain. We argue that a better insight into the decision process can be provided through a workflow model showing: the data elements that should be used to produce the final decision and how they need to be manipulated (the sequence of actions to be performed).

The main benefit of our method is that it is faster, less expensive and more objective than the usual knowledge extraction methods (such as questionnaires, interviews, etc). From a model produced using this approach, people who want a better insight into a specific real life decision (e.g. managers, students) can further extend their knowledge by reading and understanding a process performed by an expert. Our approach can also be used by professors for evaluating the progress in decision making training (by comparing the 'before' and 'after' models). Professionals interested in knowledge extraction can use our approach as an alternate knowledge extraction tool.

The structure of this paper is as follows. First, we introduce the reader to an overview of the decision mining approach. Then, Section 3 explains the concept of the PDM and discusses the related research areas. In the section on the mining approach (Section 4), we define the concepts we use and we explain the steps we follow in order to create a PDM out of user activity logs. We then show the mining algorithm and a running example. The fifth section introduces a case study and a brief discussion of an example PDM mined from the decision process of an expert user. In the last sections we provide an evaluation of the approach and the conclusions.

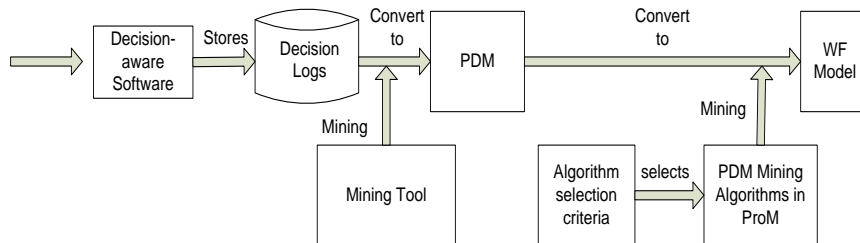
## 2. Methodology

In this methodology section we present the general overview and background needed to understand our approach. In the general overview we show how we enable the capturing of the relevant knowledge by taking a process mining perspective and making it explicit in a specific type of model, the product data model (PDM). The background section elaborates on this important notion of the PDM and briefly discusses other related issues.

### 2.1. General approach

The general approach for making the relevant knowledge for a decision process explicit is illustrated in Figure 1. First, we ask the decision makers to use our ‘decision-aware software’. This software provides the decision makers with a lot of data for a decision scenario, ranging from trivial to critical (e.g. all the data and information outputted by the information system of a company for an economic decision). The term decision-aware designates the fact that the software is built so it will [PM, 2010]:

- enable the user to perform all the mental steps towards making a decision within the boundaries of the system (mental steps are e.g. viewing a data item, comparing data items, calculating a derived data item, etc.);
- ‘force’ the user to decompose a mental pattern into basic thinking items;
- ‘force’ the user to express each basic thinking item as an interaction with the system that can be logged.



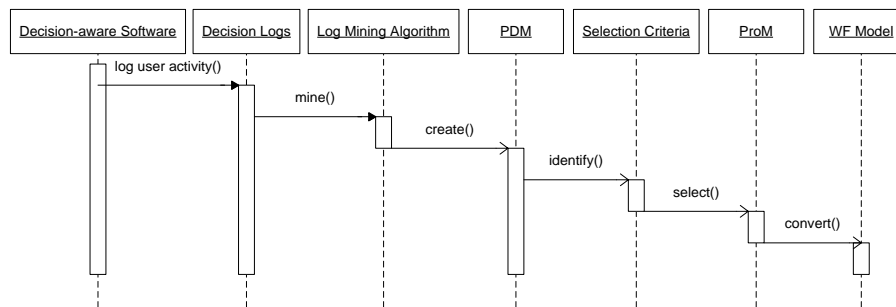
*Figure 1: General approach for decision mining*

The software stores all details of the user interaction with the system in a set of tables [PM, 2010]. For the purpose of this paper we will focus only on the data elements that are considered when a decision maker is performing a specific decision process. This is referred to as a ‘trace’ of the process. Each trace may consist of:

- basic data items available in the simulation scenario. This data is pre-set for a decision process by the moderator and can emphasize on a specific behavior that needs to be researched. We log the name and the value of the basic data items that are viewed and used in any way by the decision maker;

- data items inputted by the user. There may be data items that are not available in the simulation data but the user may consider important. Therefore, the user may type in new data and use it in deriving new items. We log for each such data item: a mandatory description and the value inputted by the user;
- derived data items calculated by the user. The derived items can be calculated based on: basic data or/and on inputted data or/and previously derived data items. We log for each such item: the source items and the operands used.
- type of interaction with the system. For example, all basic data item values are by default hidden. In order to look at the value of such an item the user must explicitly click the textbox containing that particular value (e.g. we will log the “click textbox” interaction).
- timestamp of each interaction. This allows us to order the actions of the user stored in the log.

After capturing the traces in the log, these are converted to a PDM by our decision mining tool. The PDM is represented by an XML file and can be read by ProM. After that, the PDM may be converted into a workflow process model using the available algorithms in ProM. There are currently six algorithms that can create a workflow model out of a PDM file [Vanderfeesten, 2008], each of them emphasizing on a particular view of the process. Therefore, there needs to be selection criterions for choosing one algorithm over the others that will rely on the user’s needs. A decision workflow model needs to show to the users as many insights regarding: what were the data items used and derived; how they relate to each other; and which is the sequence of usage.



*Figure 2: The process of deriving a workflow model from an activity log*

The complete process of deriving a workflow model is introduced in Figure 2. To illustrate this process, we introduce a short example.

**Example 1.** In this example, the user needs to decide on whether or not to grant a mortgage. The available data items in this particular environment are: percentage of interest (B), the annual budget to be spent on the mortgage (C), the term of the mortgage (D), the previous mortgage offer (E), the income the client is allowed to spend on paying the mortgage (F), the gross income of the client per year (G), and the credit registration (H). The output of the decision process is the value of the

maximum mortgage that needs to be typed and will be stored as item A. In order to perform this decision the following steps are taken:

- The decision maker uses the decision-aware software and considers that he needs to fundament the decision for the value of A on the values of B and C, and the difference of values F and C (F-C). He expresses this while using the software by: looking at values B, C, F and deducting C from F.
- The log outputted by the software for this particular trace consists of: the values of data items: B, C, F, and (F-C); the timestamps for each action, the type of interaction with the software (e.g. click textbox, click button), and the sign of the operand used while deriving data, etc.
- The log is mined (by using the mining algorithm introduced later in this paper), and an XML file consistent with the PDM definition (as presented in the next section) is delivered;
- The XML file can be loaded in ProM and converted to a workflow model by using an algorithm that will produce an appropriate representation.
- The model we will present back to the user (depicted as a Petri Net) will look like the one shown in Figure 3:

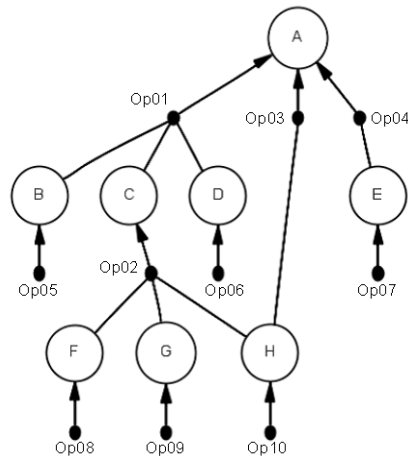
We basically argue that: a) such a model depicts clearly the mental actions of the user and their order, b) it will take us less time to create it then classic knowledge acquisition methods when applied to a large number of users and c) it can be easily understood (therefore knowledge is easier disseminated).



*Figure 3: Example of a decision process modeled as a Petri Net*

## 2.2. Background

To understand our approach, the reader should be familiar with the concept of a PDM which is explained below. The Product Data Model (PDM) is a well-known concept from the area of business process (re)design. It is the starting point for the Product-Based Workflow Design (PBWD) methodology [RLvdA, 2003], [Vanderfeesten, 2010].



**Figure 4: The PDM for the mortgage example**

A product data model describes the structure of the process of information processing to produce an informational product. It is similar to a Bill-of-Materials (BoM), [Orlicky, 1972]. The product that is described by a PDM is an informational product, e.g. a decision on an insurance claim, the allocation of a subsidy, or the approval of a loan. In a PDM the data elements that play a role in a decision and their relationships are made explicit in a graphical way.

Consider for instance the example in Figure 4. This example describes the calculation of the maximum amount of mortgage a client is able to borrow from a bank as was discussed before. The figure shows that the maximum mortgage (element A in Figure 4) is dependent either on a previous mortgage offer (E), or on the registration in the central credit register (H), or on the combination of the percentage of interest (B), the annual budget to be spent on the mortgage (C), and the term of the mortgage (D). The annual budget (C) is determined from the gross income of the client per year (G), the credit registration (H), and the percentage of the income the client is allowed to spend on paying the mortgage (F).

Data elements are depicted by circles in the PDM. For each specific case instance of the decision process a data element may have a different value (e.g. the value for data element “interest percentage” may be different for a long term loan than for a short term loan, also, the gross income of each client will be different).

The actions that are taken on the data element values are called operations and are represented by hyperarcs. In general, an operation can be of different forms, e.g. an automatic calculation, a judgment by a human or a rule-based decision.

Each operation has zero or more input data elements and produces exactly one output data element. The arcs are ‘knotted’ together when a value for all data elements is needed to execute the particular operation. Compare for instance the arcs from B, C, and D leading to A on the one hand, and the arc leading from E to A on the other hand in Figure 4. In the latter case only one data element value is needed to determine the outcome of the process, while in the case of B, C, and D all three data element values are needed to produce A. An operation is executable when a value for all of its input elements is available.



Several operations can have the same output element while having a different set of input elements. Such a situation represents alternative ways to produce a value for that output element. For example, a value for the end product A in Figure 4, can be determined in three alternative ways: (i) based on a value for E, (ii) based on a value for H, and (iii) based on values for B, C, and D. Also, a data element may be used as an input element to several operations. For instance, H is used in two operations: Op02 and Op03

The top element of the PDM, i.e. the end product, is called the root of the PDM. The leaf elements are the elements that are provided as inputs to the process. They are produced by operations with no input elements (e.g. the operations with output elements B, D, E, F, G, and H). The operations producing values for the leaf elements are denoted as leaf operations or input operations.

Note that the structure of the PDM is a network structure (it is not a simple tree), but does not contain cycles.

Current algorithms for process mining all focus on the retrieval of a process model from an event log. In our mining approach we try to mine the processing of information.

### **2.3. Related work**

This research draws on several major fields of research: workflow management (especially process mining), decision making theory and analysis, decision support systems and software simulations.

The process mining methodology is present for two decades [CW, 1998]. The work was extended to concurrent processes and tasks which were discovered out of event logs by using entropy, event type counts, periodicity, and causality. More recent approaches of business process mining aim to analyze existing event logs produced by process or workflow aware software (such as ERP, CRM, SCM, etc) [vdAH, 2002] [vdAvDHMSW, 2003]. The result of process mining is a model that reflects a real life process in an enterprise. The decision mining approach we present in this paper resembles process mining in that it aims to automatically extract and create a model, but of the mental decision making process rather than of some physical process in the enterprise. Our approach is based on the fact that the actions of a person will provide an external observer with a better understanding of a workflow than what the user says about that workflow. This assumption is also used by various researchers in process mining that rely on the historic operational data available from event logs (or audit trails or transaction logs, etc) produced by the software tools used in an enterprise (ERP, CRM, SCM, etc) rather than on the prescribed workflows modeled by experts [vdAvDHMSW, 2003], [vdAW, 2004].

The term “decision mining” was used before in [RvdA, 2006]. The mining algorithms are implemented in a plug-in called Decision Miner that is implemented in ProM Framework. This approach uses a derivation of C4.5 algorithm to build decision trees that allow analysis of choices in the decision points of a workflow. Rozinat proposes the use of Petri Nets theory in order to identify the points in which a choice was made and one or other of the branches were followed. After the decision point is found, the problem is turned into a classification problem that tries to

determine if the cases with certain properties follow specific routes. However, this is different from our approach. The difference is that in a process mining log the traces repeat as the users perform the same activities in the same order prescribed by the company's procedure or enforced by the company's software systems. In decision mining, a process is basically unique since no two persons have the same knowledge and even the same person does not necessarily follow the same thinking pattern twice. The tree like structure can be obtained only if a large number of models are aggregated. Even at this level, there are differences due to the fact that the properties determined by Rozinat that change the path followed for a decision point cannot map to the properties of the mental activities captured in the decision workflow. Rozinat's approach is focused on determining the throughput times for decision nodes in order to improve the process. Such a goal is of no consequence to our approach. Also, Rozinat's approach is not necessarily based on the value for case data. She also considers 'changes' to go into one direction or the other in a process model, while we specifically focus on the content of the decision.

Decision analysis is the discipline comprising the philosophy, theory, methodology, and practice necessary to address important decisions in a formal manner [GW, 2004]. We draw on decision analysis' documented procedures, methods, and tools used for identifying, representing, and assessing the important aspects of a decision situation. Some of our higher goals include prescribing the recommended course of action in a decisional situation by applying some theorems and algorithms to create and exploit a well-formed representation of the decision process. We also aim to create formal representations of decisions and translate them into new insights and recommendations for the decision maker. Our immediate goals include identifying and explaining various aspects of actual decision making processes (using tools such as graphical representations, variables, uncertainty, etc)

The class of systems which aims to provide the user with all the necessary data and information in order to help him make better decisions is the class of decision support systems (DSS). A brief overview of the DSS research area is available in [Power, 2004]. Current research is focused on adding more "intelligence" to the system and integrating the DSS within the business intelligence systems in the enterprises [TSD, 2010]. In order to create a successful decision-aware system, we need to join a DSS with a virtual environment in order to provide the user with the best decision experience in a simulated decision environment. Some of the questions that are already addressed in the DSS area, we also need to consider while building the decision-aware software: "how should the data be presented to the user?", "which data should be available?", "what tools should the decision maker use in the decision process?".

### **3. Approach**

In this section we will introduce the formal definition of the key concept of PDM in the first sub-section. Then, we will describe the steps we propose in order to mine the

PDM model out of the activity traces. In the second sub-section we will demonstrate the theoretical framework and the proposed approach by a running example.

### 3.1. Definitions

To be able to specify our decision mining algorithm we need to define a number of concepts. First of all, we look at the definition of the PDM, which is slightly modified and simplified from the general definition in [Vanderfeesten, 2008] in order to make it fit for our decision mining purpose.

**Definition 1 (PDM).** A PDM is a tuple  $(D;O;T)$  with:

- $D$ : the set of data elements,  $D = BD \cup DD \cup ID$ , with
  - $BD$  the set of leaf data elements
  - $DD$  the set of derived data elements
  - $ID$  the set of data elements inputted by the user
- $O \subseteq D \times P(D)$ : the set of operations on the data elements. Each operation,  $o = (d; ds)$ :
  - has one output element  $d \in DD$  and
  - has a set of zero or more input elements  $ds \subseteq D$
- $T: O \rightarrow \mathbb{R}$ : the partial function that specifies the amount of time required to execute an operation from  $O$ , i.e. the time required to produce the element  $d$  based on the elements  $ds$  using  $ao$ .
- $D$  and  $O$  form a hypergraph  $H = (D;O)$  such that its structure graph is connected and acyclic.

The PDM of Figure 4, contains six leaf elements:  $B, D, E, F, G, H \in BD$ . The leaf element  $B$  is for instance produced by an operation  $op_{05} = (B; \emptyset)$ . There are also two derived data elements:  $A, C \in DD$ . And the root element, data element  $A$ , is produced by operation  $op_{01} = (A; \{B, C, D\})$ .

Our approach is based on our ability to extract data from the logs of the software and present them as a PDM. This is done in three major steps:

A) parse the logs and output a XML file,

A1) export the logs from the decision-aware tool;

A2) filter the logs for just one trace. This is based on the Process Instance ID;

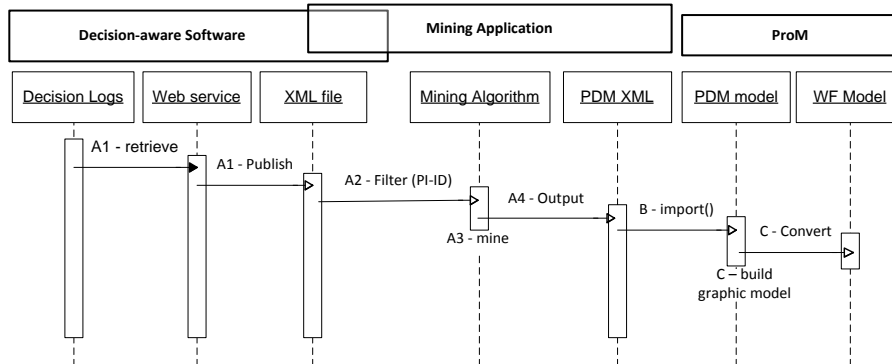
A3) run the mining algorithm on the individual trace so that relevant information is extracted from the logs;

A4) input the data sets into the specific structure of the PDM XML file;

B) import the XML file into ProM Framework

C) build the PDM and the workflow model.

A summary for the steps we introduce for our approach above is depicted in Figure 5:



*Figure 5: The sequence of steps for transforming activity data into a workflow model*

Activity A1 is performed by a web-service included in the decision aware system. It allows the mining application to retrieve the necessary data (as an XML file). Depending on the context, Activity A2 can be performed by the decision-aware system (if a user wants to build the model right after he finished performing the decision process) or by the mining application (if a researcher using our approach wants to build one process model out of a log containing multiple traces). Activity A3 is performed by the stand-alone mining application. For a better understanding we will introduce the algorithm implemented in the application as pseudo-code in the next sub-section. The input data for the algorithm is one trace in the activity logs outputted by the decision-aware system formatted as one XML file (see Activity A1). The mining application also performs Activity A4 and outputs a PDM specific XML file that contains all the elements introduced in Definition 1. So far, this file needs to be manually uploaded into ProM (Activity B). The ProM plug-in creates the PDM graphical representation and the various workflow models (Activity C) [Vanderfeesten, 2008].

### 3.2. The decision mining approach

The main concern of this sub-section is introducing the reader into how the XML file containing the structure of the PDM is produced (activity A). We will introduce the algorithm implemented in the mining application and then a running example that will provide a better understanding of how the PDM can be created based on the logged behavior.

The mining approach we introduce in this paper focuses on the data perspective of the decision process. Therefore, we are concerned on extracting two things: the basic data items used by the decision maker and how those items are derived and combined with each other. If the decision maker intends to use a derived data item in the decision process, he first needs to calculate it. The calculation steps are not always visible to the outside observer because they are most of the times done mentally. So far, the only methods, that we are aware of, broadly used for extracting the mental

process are the interviews (the user is asked to report what he did something) and “thinking-aloud” (the user is asked to report while he is doing something).

The mining algorithm implemented in the mining application performs mining based on the following logic:

```

Create leaf_node (BD ∪ ID) set
Create derived_data_element (DD) set
Create root_node (RT) set
Create operation (O) set
Create operation_data_elements set
Start top
Do case for each record
    Case Find_click_textbox() = True
        If textbox not in leaf_node set
            Create a new leaf node
            Label leaf node with Name Field value
            Add to leaf_node set
        Else
            EndIf
    Case Find_operation() = True
        Create new operation
        Add it to operation set
        Create a new derived data item
        Name it (autonumber with letters)
        Place it in the derived_data_element set
        For each item in leaf_node and derived_data_element sets
            Do used_in_operation()
                Add to operation_data_elements: (name of
                current operation, name of current node element
                as output, all data items (leaf nodes and derived
                data elements) found by used_in_operation as
                input)
        Endcase

Do Find_edit_textbox()
Place value in Name Field in Root_node set

Create new operation
Add it to operation set
Do for each element in leaf_nodes
Add to operation_data_elements: (new name for operation, name of
current leaf_node as output, Null as input)
Search Leaf_node set and Derived_data_element set
For each item not used in deriving other data elements
    Create new operation
    Add it to operation set
    Add to operation_data_elements: (name of current
    operation, Root_node as output, empty as input)

EndIf

```

Some explanations on the functions used in the algorithm introduced above:

- Find\_click\_textbox() – looks in WFMElt Field for the “click textbox” values
- Find\_operation() – looks in Name Field for the values starting with “=” character
- Used\_in\_operation() – looks inside the records in Name Field starting with “=” character and extracts: all the data labels followed by one of the operation signs (+, -, \*, /) and the operation signs. If there is an “=” character inside the

expression the function extracts the items between the parenthesis that contain the “=” sign and searches the node\_elements string for the derived data item name. This function produces an ordered set of items (either a leaf node or a node element) and an ordered set of signs linked to the previous set by the ordering inside the set. For example, in Table 1 at record 6 there is the expression: =XA+XB. This function extracts the two input elements XA and XB and names the result A. It outputs, according to Definition 1, the string (A, {XA, XB}) . In table 1 at record 11 there is the expression: “=(XA+XB=)/XC”. This function detects the second equal sign in the expression and therefore recognizes the expression in the parenthesis already exists and matches it to A so the operation is reduced to “=A/XC”. The strig outputted for the second operation’s data elements is (B, {A, XC}).

In the remainder of this sub section, we will focus on how we can explicitly show as a PDM model what are the data items and the operations performed by the user while calculating a derived data element. In the context of the decision process this is important because we need to show how this derived value fits into the overall decision process. For a short example, we suppose the user needs to know the result of the formula (as a naming convention, we use X in front of any basic data item and we assign sequential letters for any calculated item):

$$(XA + XB) / XC = ? . \tag{1}$$

Where: XA = 1000, XB = 500 and XC = 5.

When calculating such a result the mental actions performed by the user are:

- a) check for the value of XA, then remember it for the calculation,
- b) check for the value of XB, then remember it for the calculation,
- c) calculate the result of the addition,
- d) check for the value of XC, then remember it for the calculation,
- e) calculate the final result by dividing the result of the previous addition to the value of XC.

If those calculations steps are performed within the decision-aware system we will be able to generate a log of all the steps performed by the user while performing the calculation. This is how we make explicit a part of the knowledge employed by the user. The log sample, generated by the interaction of the user with the decision-aware system, for calculating the formula introduced above is:

**Table 1. Log explicitly depicting mental calculation steps**

Timestamp	WFMElt	Name	Data attributes
Time 1	click textbox	XA	1000
Time 2	click button	Add_XA	XA
Time 3	click button	plus	+
Time 4	click textbox	XB	500
Time 5	click button	Add_XB	XB
Time 6	click button	=XA+XB	1500
Time 7	click button	Add_(XA+XB=)	(XA+XB=)
Time 8	click button	divide	/
Time 9	click textbox	XC	5

Time 10	click button	Add_XC	XC
Time 11	click button	=(XA+XB=)/XC	300
Time 12	edit textbox	XD	300

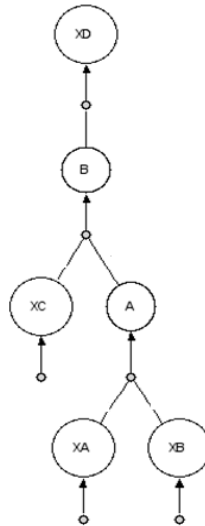
The log actually shows all the calculation steps that are now explicitly performed by the user as a sequence of interactions with the decision aware system. Assuming this to be a complete trace in which the user's goal is only to produce the result of the formula, the final step that needs to be performed by the user within the system is to write down the value of the calculation (edit textbox XD in record 12).

By running the mining algorithm we can produce the following PDM output consistent with Definition 1:

*Data element (D):* {XA, XB, XC, XD, A, B}  
*Leaf node set (BD):* {XA, XB, XC}  
*Root element* XD  
*Derived data set (DD):* {A, B}  
*Operation set (O):* {op1, op2, op3, op4, op5, op6}  
 where: (op1, A, {XA, XB}), (op2, B, {A, XC}),  
 (op3, B, XD), (op4,  $\emptyset$ , XA), (op5,  $\emptyset$ , XB),  
 (op6,  $\emptyset$ , XC),

Based on the above sets, we can create the XML input file that can be imported to ProM Framework for creating the graphical representation of the PDM model. The XML file produced using the sets outputted by the mining algorithms is presented in Figure 11 .

The approach used to create the PDM graphical representation is the one presented in [Vanderfeesten, 2008]. The order of the elements in the PDM input data is of no consequence. The PDM is constructed starting from the root node in a top down manner [Vanderfeesten, 2008]. By feeding the PDM-XML file into ProM Framework we created the PDM model shown in Figure 6.



**Figure 6: PDM model built on the mined XML**

## 4. Case Study

This section introduces several case studies performed while testing our approach.

The first sub-section shows how the decision process introduced earlier, as the running example, was performed using the software implementing our approach. The goal of this sub-section is to get the reader acquainted with the way in which the user interacts with the decision-aware system. This is important because the whole approach relies on the assumption that we are able to capture a mental decision process. As mentioned earlier, the interaction with the system aims to force the user to break up complex and implicit mental actions into atomic items that are performed explicitly as a sequence of actions within the system. Besides the actual interaction with the software we continue to illustrate, using this example, the entire process of extracting the instance PDM.

The second sub-section introduces the decision mining process performed for a trace produced by several expert users. Basically, what we introduced in the first sub-section as a short example is performed by an expert decision-maker according to his expertise. We will not emphasize on the interactions with the software because it basically works as in the running example. Instead we will better introduce the reader to the loan contracting decision set-up and will introduce several PDM models that will underline the fact that, for the same decision, the approaches of experts can be extremely diversified.

The last two sub-sections introduce several models produced by using our approach for several intermediate traces and one beginner trace. A comparative discussion is provided based on the expert, intermediate and beginner traces.

### 4.1. Usage Scenario Based on the Running example

This section introduces a complete example of how an instance PDM can be derived. We first show how the user interacts with the decision-aware software that presents the loan contracting scenario data. Then we show the logs outputted and the PDM XML created by the mining algorithm. In the end, we show the workflow models created in ProM.

We will shortly show how the running example introduced above is actually performed in practice, by using a decision-aware system and the mining algorithm. We will instantiate XA as accounts receivables, XB as accounts payables, XC as cash and collateral cash; and XD the total amount to be invested. The user decides, based only on those four data items, how much money he needs to borrow from the bank (the root element, previously noted XD in the running example, is now named RT so it can be easily distinguished from the actual data items in the model).

In order to produce the decision process model, the user first needs to interact with the decision-aware software. The goal of those first actions is to derive data based on the data items provided in the simulation scenario. The sequence of interaction is shown in the next three screenshots (Figure 7, Figure 8, and Figure 9):



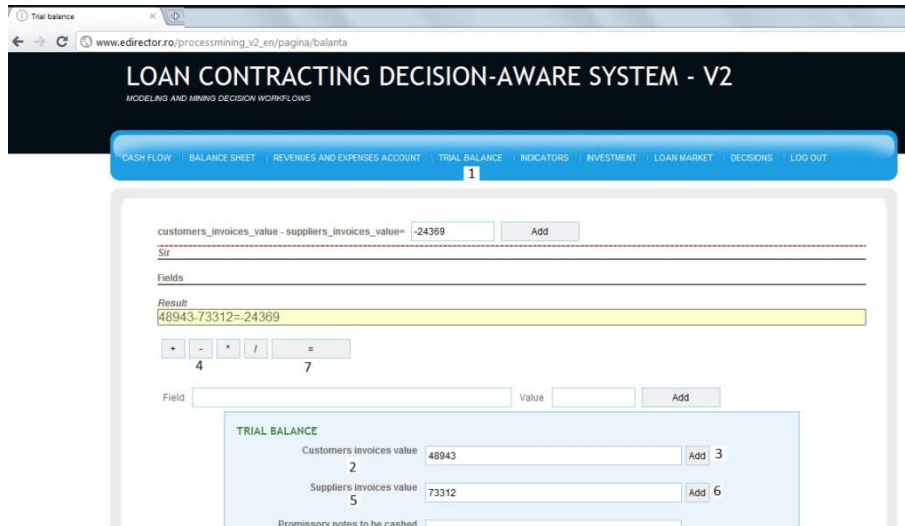


Figure 7: Sequence of interaction for calculating the 'future cash need' as the difference between 'accounts payable' and 'accounts receivable'.

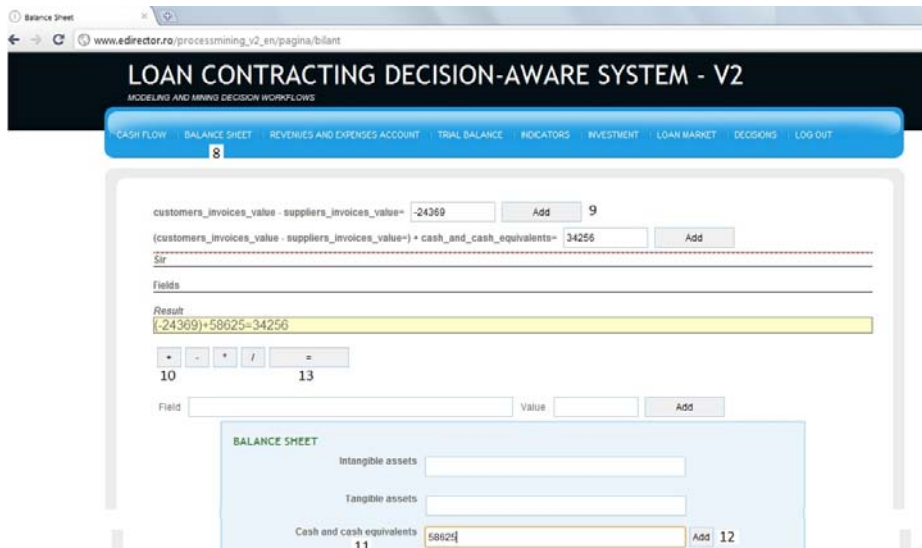
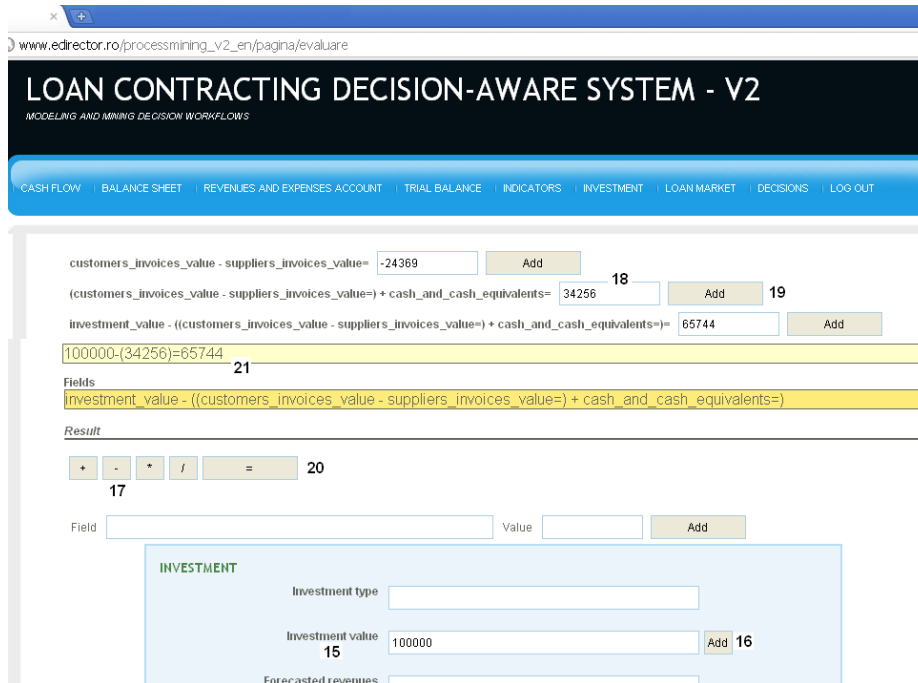
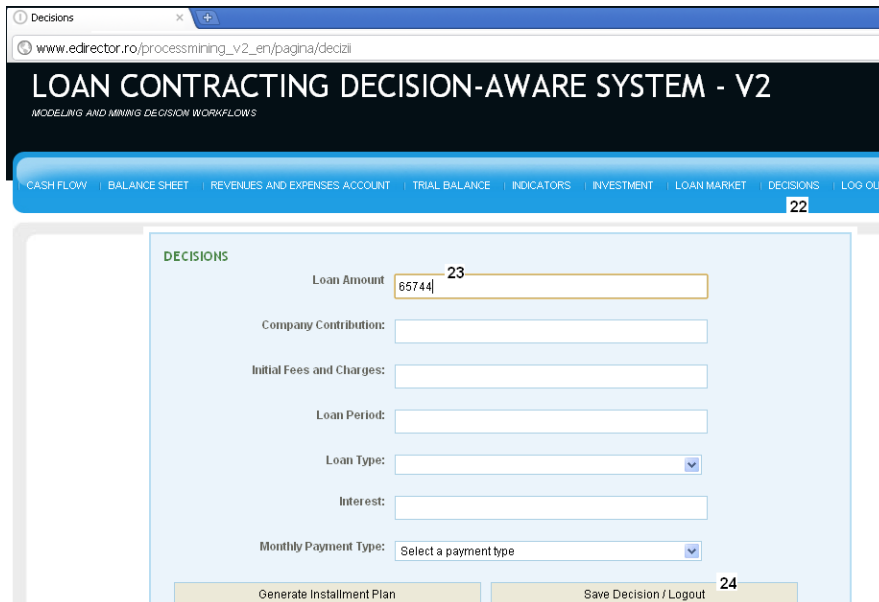


Figure 8: Sequence of interaction for calculating the 'cash available for investment' as the sum of 'future cash need' and 'cash and cash equivalents'.



**Figure 9: Sequence of interaction for calculating the 'amount needed for investment' as the difference between the 'value of investment' and the 'cash available for investment'**



**Figure 10: Final step of the interaction with the decision-aware system: typing a value for the loan amount.**

```

<export_id>26</export_id>
<PI ID>26</PI ID>
<Originator>test</Originator>
<Timestamp>01/22/2011 19:13:51.736</Timestamp>
<WFMEIT>click menu item</WFMEIT>
<ATE ID>769</ATE ID>
<DAATE_Name>Add_investment_value</DAATE_Name>
<Description>loan_value_decision</Description>
<export_id>26</export_id>
<PI ID>26</PI ID>
<Originator>test</Originator>
<Timestamp>01/22/2011 19:13:23.658</Timestamp>
<WFMEIT>click menu item</WFMEIT>
<ATE ID>763</ATE ID>
<DAATE_Name>balance_sheet</DAATE_Name>
<Description>loan_value_decision</Description>
<export_id>26</export_id>

```

Tag name	PI ID	Orig	Timestamp	WFMEIT	ATE	DAATE_Name	Descrpt...	DA
export_26	test	01/22/2011 19:12:3	click menu item	754	trial_balance	loan_value...		
export_26	test	01/22/2011 19:12:4	click textbox	755	customers_invoices_value	loan_value...	48943	
export_26	test	01/22/2011 19:12:4	click button	756	Add_customers_invoices_value	loan_value...		
export_26	test	01/22/2011 19:12:4	click button	757	minus	loan_value...		
export_26	test	01/22/2011 19:12:4	click textbox	758	suppliers_invoices_value	loan_value...	73312	
export_26	test	01/22/2011 19:12:5	click button	759	Add_suppliers_invoices_value	loan_value...		
export_26	test	01/22/2011 19:12:5	click button	760	=customers_invoices_value - suppliers_invoices_value	loan_value...		
export_26	test	01/22/2011 19:13:0	click button	761	Add_(customers_invoices_value - suppliers_invoices_value)	loan_value...		
export_26	test	01/22/2011 19:13:1	click button	762	plus	loan_value...		
export_26	test	01/22/2011 19:13:2	click menu item	763	balance_sheet	loan_value...		
export_26	test	01/22/2011 19:13:2	click textbox	764	cash_and_cash_equivalents	loan_value...	50625	
export_26	test	01/22/2011 19:13:2	click button	765	Add_cash_and_cash_equivalents	loan_value...		
export_26	test	01/22/2011 19:13:3	click button	766	=(customers_invoices_value - suppliers_invoices_value) + cash_and_cash_equivalents	loan_value...		
export_26	test	01/22/2011 19:13:3	click menu item	767	evaluate_investite	loan_value...		
export_26	test	01/22/2011 19:13:4	click textbox	768	investment_value	loan_value...	100000	
export_26	test	01/22/2011 19:13:5	click button	769	Add_investment_value	loan_value...		
export_26	test	01/22/2011 19:13:5	click button	770	minus	loan_value...		
export_26	test	01/22/2011 19:13:5	click button	771	Add_((customers_invoices_value - suppliers_invoices_value) + cash_and_cash_equival...	loan_value...		
export_26	test	01/22/2011 19:14:2	click button	772	=investment_value - ((customers_invoices_value - suppliers_invoices_value) + cash_and...	loan_value...		
export_26	test	01/22/2011 19:24:1	click menu item	773	Decizi	loan_value...		
export_26	test	01/22/2011 19:24:2	edit textbox	774	loan_amount	loan_value...	65744	
export_26	test	01/22/2011 19:26:0	click button	775	decision_save	loan_value...		

Figure 11: The XML file containing the user activity logs outputted by the decision-aware software system

```

<Name>Running Example Log</Name>
<DataElement DataElementID="XA"/>
<DataElement DataElementID="XB"/>
<DataElement DataElementID="XC"/>
<DataElement DataElementID="XD"/>
<DataElement DataElementID="A"/>
<DataElement DataElementID="B"/>
<DataElement DataElementID="RT"/>
<Resource ResourceID="w1"/>
<Operation OperationID="Op01"/>
  <Input />
    <DataElementRef>XA</DataElementRef>
    <DataElementRef>XB</DataElementRef>
  </Input>
  <Output />
    <DataElementRef>A</DataElementRef>
  </Output>
  <ResourceRef>w1</ResourceRef>
  <Condition>true</Condition>
</Operation>
<Operation OperationID="Op02"/>
  <Input />
    <DataElementRef>A</DataElementRef>
    <DataElementRef>XC</DataElementRef>
  </Input>
  <Output />
    <DataElementRef>B</DataElementRef>
  </Output>
  <ResourceRef>w1</ResourceRef>
  <Condition>true</Condition>
</Operation>
<Operation OperationID="Op03"/>
  <Input />
    <DataElementRef>B</DataElementRef>

```

Tag name	Text	DataElementID	ResourceID	OperationID	Input	Output	ResourceRef	Condition
Name	Running Example Log							
DataElement		XA						
DataElement		XB						
DataElement		XC						
DataElement		XD						
DataElement		A						
DataElement		B						
DataElement		RT						
Resource			w1					
Operation				Op01	Input	Output	w1	true
Operation				Op02	Input	Output	w1	true
Operation				Op03	Input	Output	w1	true
Operation				Op04	Input	Output	w1	true
Operation				Op05	Input	Output	w1	true
Operation				Op06	Input	Output	w1	true
RootElementRef	RT							

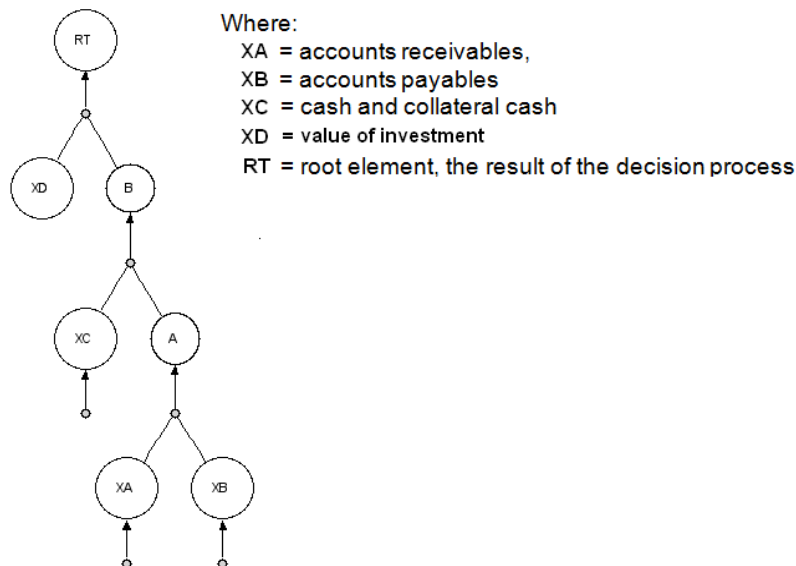
Figure 12: PDM-XML format that can be imported into ProM

After choosing a decision alternative (which for this particular situation is to decide the amount of money to be loaned), the user needs to input the decision outcome (Loan Amount Textbox) as shown in Figure 10.

After the user has saved the decision and logs out of the system, the decision-aware system will output an XML file containing the activity log performed for this particular trace. The file is presented in the Figure 11.

The last step is the automatic conversion of the activity log XML into a PDM XML file. Basically, the decision-aware system allows access to the log data through a web-service. The mining algorithm performs the conversion and outputs the data needed for building the PDM specific XML. By applying the mining algorithm on the running example XML file, the following PDM XML in Figure 12 was outputted.

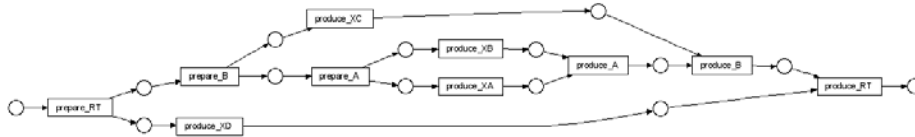
So far, the PDM file is uploaded manually into ProM 5.2. The PDM model is constructed and can be converted to workflow models. For the running example, we created the PDM model that is shown in Figure 13.



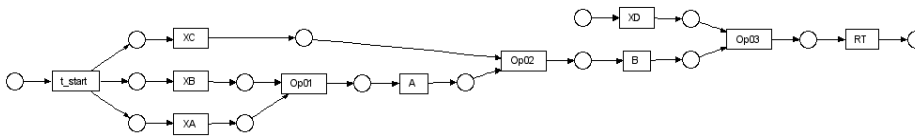
**Figure 13: PDM model created based on the user interaction with the decision-aware software system**

The generated PDM model can be automatically converted to a workflow model by using a number of seven different algorithms [Vanderfeesten, 2008]. We show two of the most meaningful and easily readable decision workflow models in Figure 14 and Figure 15. While looking at the workflow model in Figure 14 one can read:

- prepare to input the value of the loan which is based on a calculation. You first need to calculate a value based on XA (accounts receivables) and XB (accounts payables) and then aggregate it to XC (cash and collateral cash).
- once you have the value of the calculation aggregate it with the value of XD (the value of the investment) and you can produce the final result.



**Figure 14: Decision workflow model produced using algorithm Alpha in ProM**



**Figure 15: Decision workflow model produced using algorithm Charlie in ProM**

## 4.2. The Process Instance of an Expert Decision Maker

In this sub-section of the paper we are introducing the loan contracting decision process as performed by several expert decision makers we used during the experiments. The goal of this sub-section is to underline the fact that the experts were exposed to the same decision scenario data. However, using the PDMs for each user, we can show that the decision process of those experts is not following the same patterns.

The user of the decision-aware system is playing the role of a decision maker in an enterprise that already decided to make an investment. Since the company doesn't have enough money available for the investment the manager is faced with the decision of contracting a loan. The user needs to make a decision by choosing one alternative regarding: the loan value, the loan period, the loan type, and the installment type. For saving his decision, the user is required to write down some values (for loan value and loan period) and select one of the available choices (e.g. for loan type there are 6 choices and for installment type, 2) of the decision variables. The user needs to make all the decisions based only on the scenario data presented in the software and cannot update any data item. He is allowed to input additional data elements, but once an element is added it also cannot be updated.

The first part of this sub-section walks the reader through some of the actions performed by the decision maker in a real experiment trace performed by the expert. The aim is to get the reader acquainted with the way the expert user interacted with the system. In the second part of the sub-section we show the log sequence generated by the particular process instance. In the end, we show the entire PDM generated by this particular trace, we outline how it is directly extracted from the logs and provide some discussion regarding the decision process now depicted as a model.

#### 4.2.1. User interaction with the decision-aware system

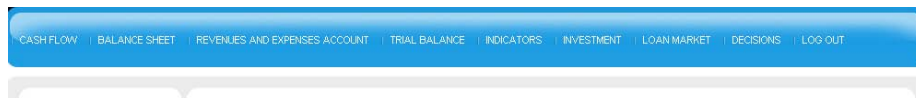
Basically, the user needs to go through all the relevant scenario data and derive new information from it. In order to keep the example simple, we asked the expert to decide only on the loan value and ignore the other aspects of a complete financing decision (the loan period, the loan type, and the installment type). Therefore, the goal is to fill-in one textbox (loan value) with the value the expert considers appropriate.

First, the user needs to log-in and select the decision he intends to make from the available options. The complete decision choice actually requires the user to fill-in all decision textboxes, while the other options allow him to concentrate only on one sub-decision. There are three types of user accounts according to the knowledge level of the decision maker: beginner (e.g. username = user1), intermediate (e.g. username = intermediate1) and expert (e.g. username = expert1). For this instance we have an expert user performing a sub-decision.

The scenario data is divided into several windows that are accessible through the top menu (Figure 16). The first five allow the user to research the financial position of the enterprise. The first four menus refer the user to the components of the annual financial statements while the fifth shows several indicators calculated on the financial data in the scenario. The Investment menu shows the data related to the investment to be financed. The Loan Market menu allows the user to look at detailed data regarding all the six loans available on the market. Finally, the decision maker needs to fill in the chosen alternative in the textboxes in the Decisions menu.

The logging is performed so that:

- a) when a page is displayed all text boxes are empty,
- b) if the user wants to see a certain data item he needs to click on the appropriate text box and the value is shown,
- c) the system logs, for the clicked item (in the tables of the database) information such as textbox's name, the value displayed in it, the timestamp, etc.



*Figure 16: All pages containing information on the decision scenario*

#### 4.2.2. Activity log outputted by the expert trace

The user interaction with the decision-aware software will be stored in five tables. Four of them are the ones required by ProM Import Tool and the fifth stores the data items inputted as decision results. The ER diagram is shown in Figure 17. A query showing the log entries generated by this particular decision process sequence is introduced in Figure 18. It can easily be observed that all the actions performed while interacting with the decision-aware software are present in the log.

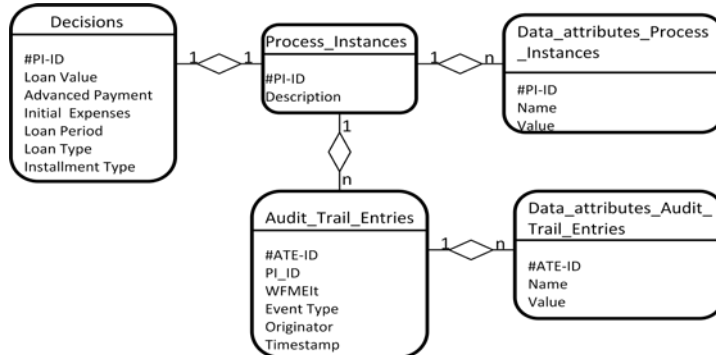


Figure 17: ER diagram of the database storing the activity logs

#### 4.2.3. Loan Value Decision PDM mined from expert trace

The PDM model mined from the first expert's log for the partial decision process (the expert was requested to decide only over the value of the loan) is shown in Figure 19. The first observation to be made on the PDM model is that it is well structured and consistent. There are also several basic data items that are used as inputs for multiple derived data items, thus making the model a network. The root element is connected with a limited number of basic data items and derived data items. This denotes careful consideration on how the actual decision value is derived.

PI-ID	Description	Value	Timestamp	ATE-ID	WFMEIt	Name	Value
33	loan_value_decision	test	02/14/2011 04:03:51:920	1046	click textbox	customers_invoices_value	48943
33	loan value decision	test	02/14/2011 04:03:53:455	1047	click buton	Add customers_invoices_value	customers_invoices_value
33	loan_value_decision	test	02/14/2011 04:03:56:550	1048	click buton	minus	-
33	loan_value_decision	test	02/14/2011 04:03:57:205	1049	click textbox	suppliers_invoices_value	73312
33	loan_value_decision	test	02/14/2011 04:03:59:447	1050	click buton	Add_suppliers_invoices_value	suppliers_invoices_value
33	loan_value_decision	test	02/14/2011 04:04:02:209	1051	click buton	=customers_invoices_value - suppliers_invoices_value	-24369
33	loan_value_decision	test	02/14/2011 04:04:07:165	1052	click menu item	cash_flow	cash_flow

Figure 18: Query showing partial log

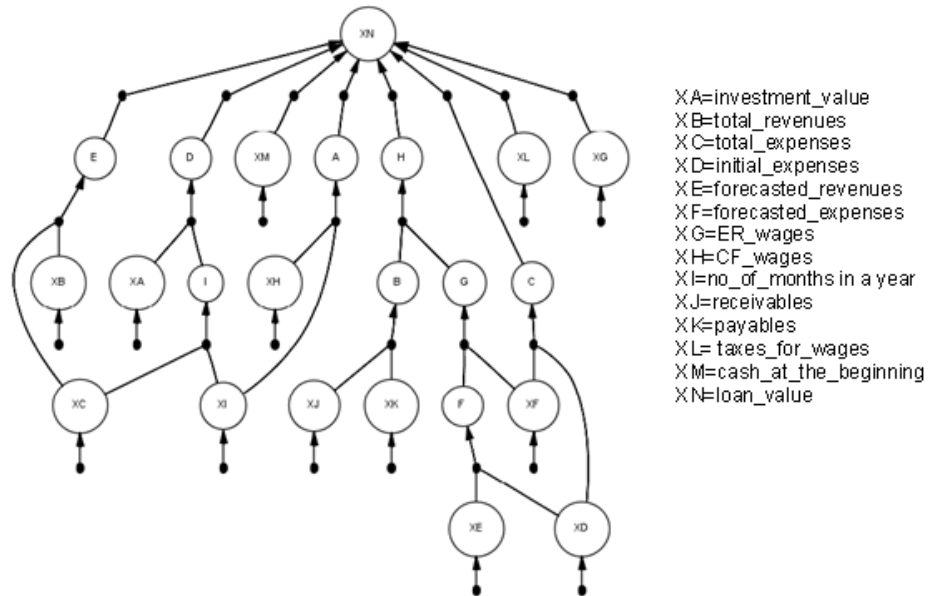


Figure 19: PDM model for an expert trace

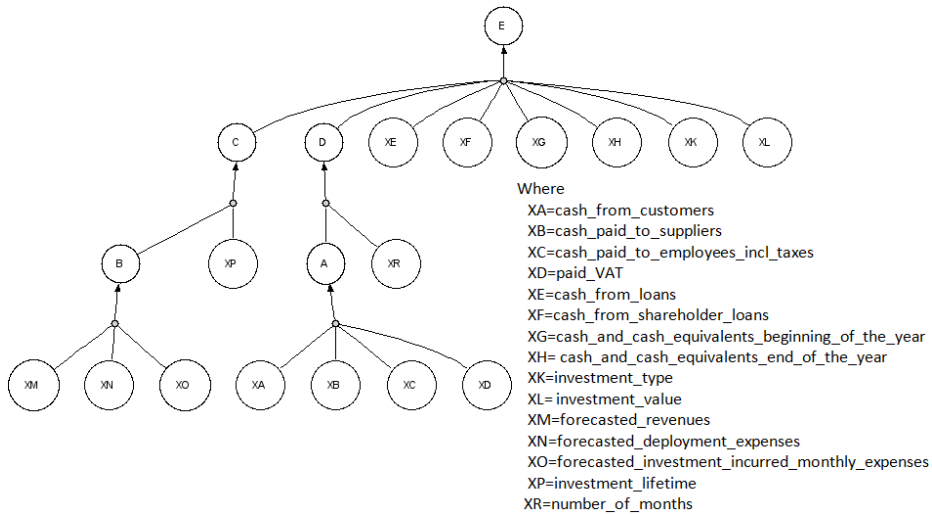
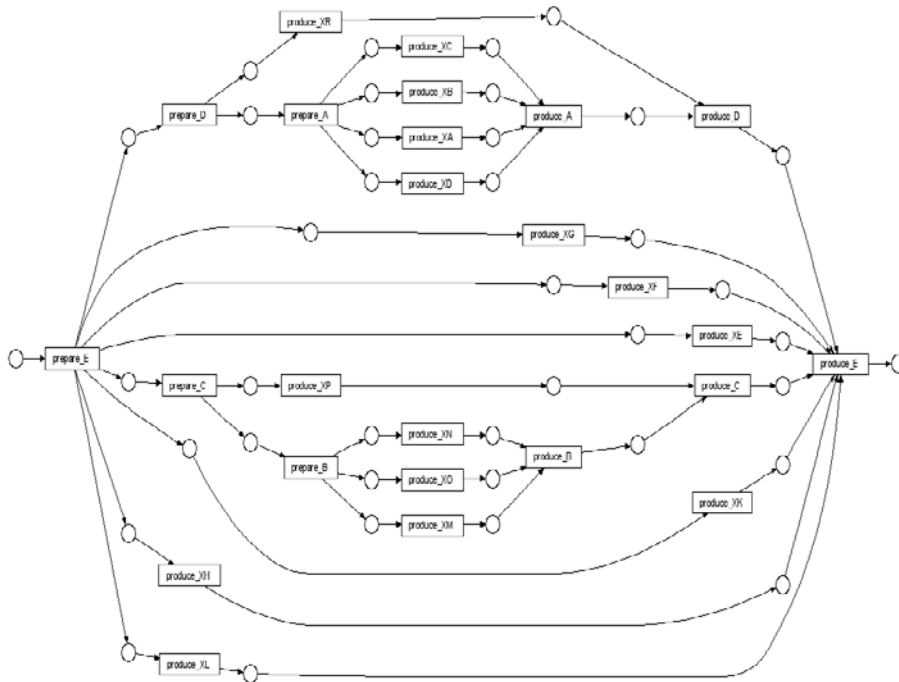


Figure 20: The PDM for a second expert trace



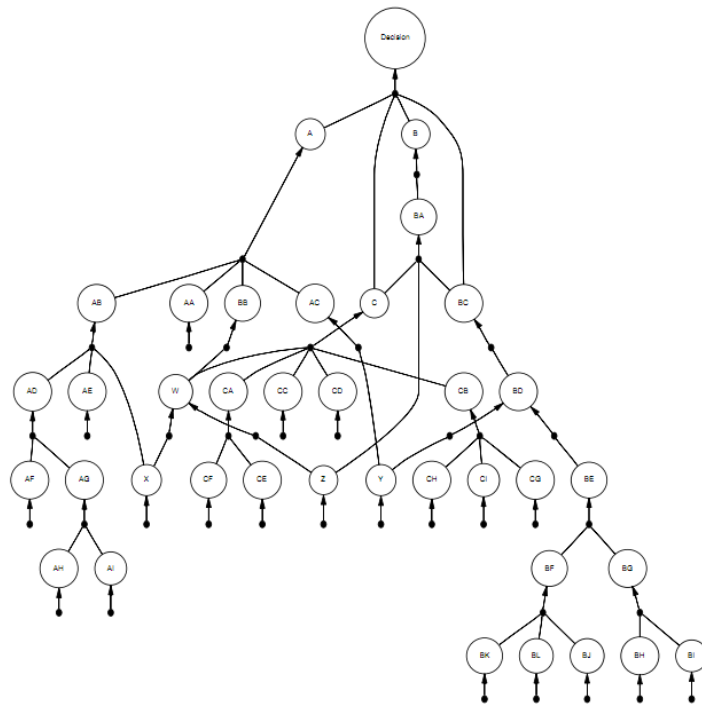
If one compares the models in Figure 19 and Figure 20 can easily observe the second PDM model is quite simplistic. There are only four derived data items. This reveals the fact that the user did not try to determine an exact value for the loan but rather went for a broader view. The fact that the root element is directly linked to a lot of basic data items is also meaningful. It reveals the fact that the user considers those items important but he doesn't link them to the result in a specific, rigorous manner. Empirically, we can state that, for this particular trace, there is not a very strong link between the decision and the data items that it is based on.

As stated in sub-section 3.2 the PDM can be converted into a workflow model depicted as a Petri Net [Vanderfeesten, 2008]. The model in Figure 20 is converted using algorithm Alpha (Figure 21) so that the actions required to create the final output are now depicted.

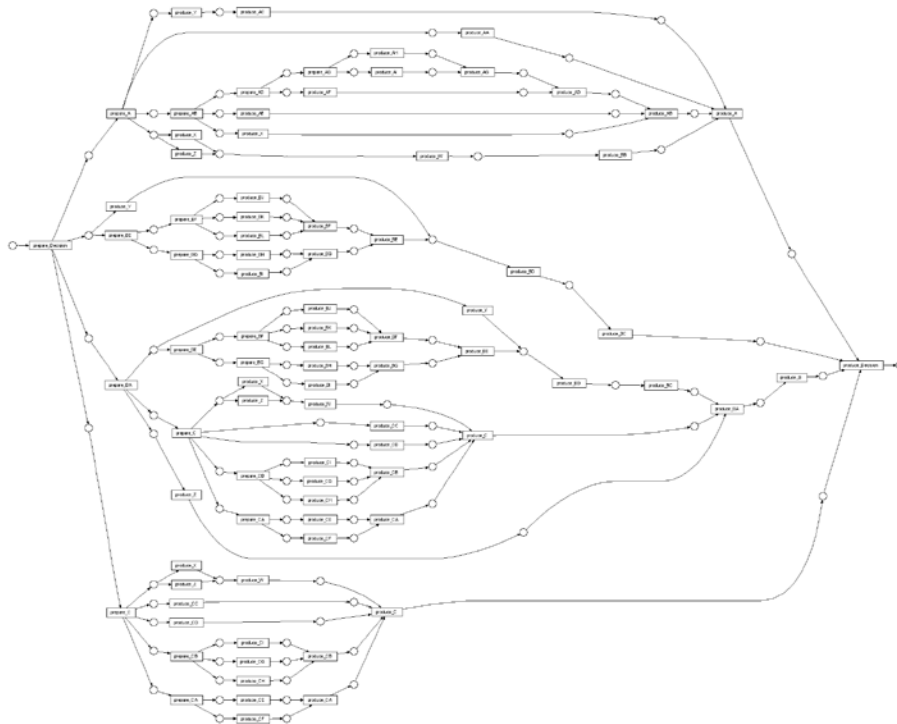


**Figure 21: Workflow based on the PDM model built using algorithm Alpha**

A model of an expert trace for the complete loan decision (the user was asked to decide over all issues concerning the loan like loan value, advanced payment value, loan period, the type of the loan, and the type of the installment) is presented in Figure 22. By comparing it to the models in Figure 19 and Figure 20 one can easily observe it includes more items and is more complex. However, it is quite readable and communicates easily how the expert decided over the problem he faced. The process model depicted as a Petri net is introduced in Figure 23.



**Figure 22: PDM model for the complete loan contracting decision**



**Figure 23: Workflow model of a complete loan contracting decision built using algorithm Alpha**

### 4.3. Process Instances Examples of Intermediate Decision Makers

This subsection introduces four of the intermediate PDM models mined after the experimental evaluation which involved second year master students enrolled in Audit and Business Information Systems curricula. The models introduced here cover a sub-decision. The users were asked to decide just on the amount to be loaned and disregard the other aspect (such as loan period, loan type, etc.). We aim to show to the reader that by simply looking at the model for a relatively simple sub-decision the processes are very different.

Another goal of this sub-section is to introduce the reader to the models that provide the basis for the discussion in the next sub-section.

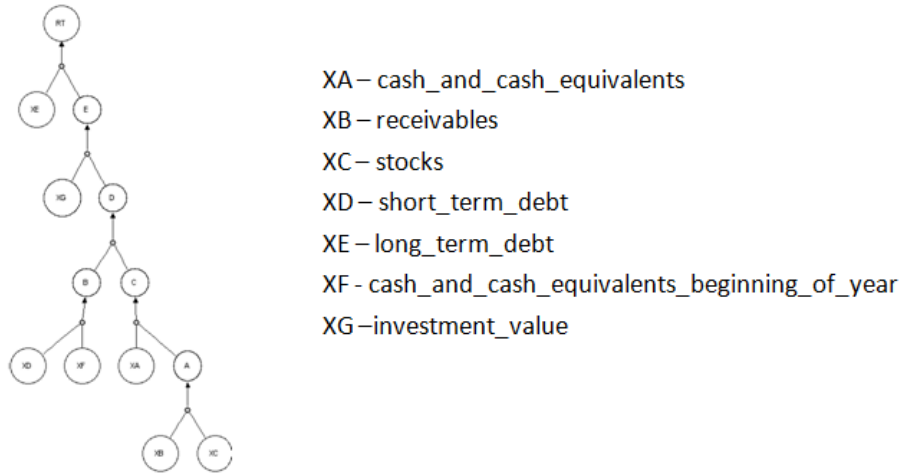


Figure 24: PDM model of intermediate decision maker

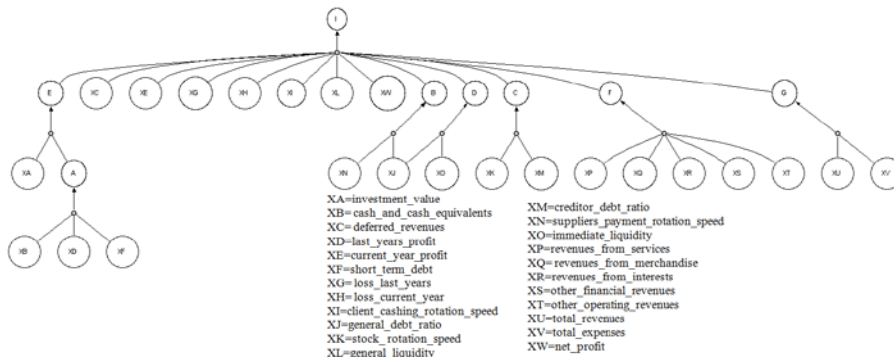


Figure 25: PDM model of intermediate decision maker

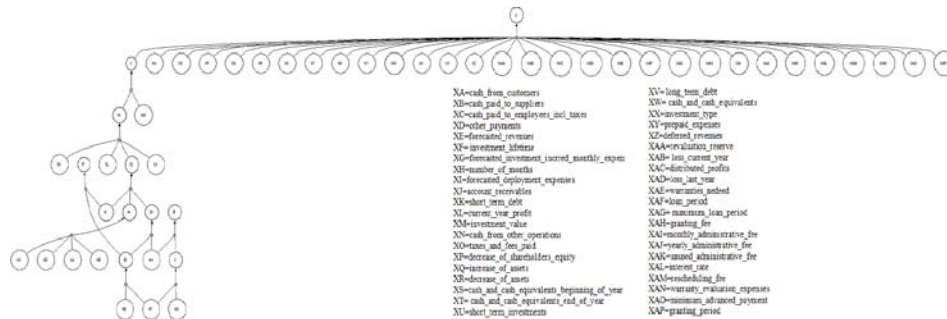


Figure 26: PDM model of intermediate decision maker

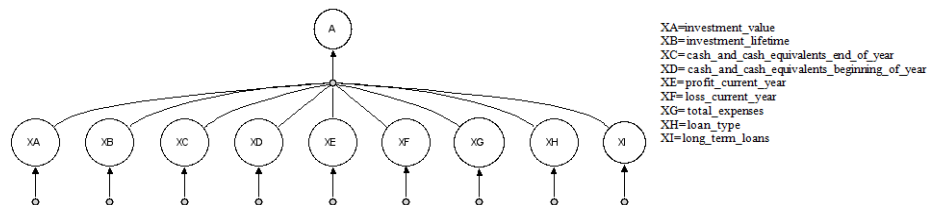


Figure 27: PDM model of beginner decision maker

#### 4.4. Discussion on Expert, Intermediate and Beginner PDM models

This discussion is based on the two expert (Figure 19 and Figure 20), the three intermediate (Figure 24, Figure 25 and Figure 26), and one beginner trace (Figure 27). A quick visual inspection reveals that the expert model involves a larger number of data items (both basic and derived) and is more consistent than the intermediate models. Only by this quick view one can deduce that the expert models exhibits more knowledge regarding the researched loan contracting decision.

By looking only at the sub-decision expert traces one can notice that there is about the same number of basic data items (13 for the model in Figure 19 and 15 for Figure 20). Out of the total basic data elements there are 6 that show up in both traces (thus making them from this point of view above 40% similar).

If we expand the analysis to both expert and intermediate sub-decision traces, one can notice that all of the decision makers used the investment value to find out the value of the loan by decreasing some expenses (e.g. total expenses (Figure 19, Figure 20), short term debts (Figure 25, Figure 26).

It is interesting to notice that all the intermediate decision makers used the short term debt figure while none of the experts did.

Five users showed interest in cash at the beginning of the year (Figure 19, Figure 20, Figure 24, Figure 26, and Figure 27), while only three used cash at the end of the year (Figure 20, Figure 26, and Figure 27). However, in none of the traces there is an operation that uses those items (it would make more sense to find out the difference between the two because it shows how much money the enterprise produced rather than just know the actual figures).

When looking at the operations performed by the users, a common operation that can be noticed in two of the traces is the difference between total revenues and total expenses (Figure 19, Figure 25). Another operation that shows up in two of the studied traces (Figure 20, and Figure 26) is the difference between cash from customers and cash paid to suppliers.

One can conclude (after performing a simple visual inspection of the sample models) that the important data items to be considered, when determining the amount of money to be loaned, are: investment value, investment lifetime, cash at the beginning, forecasted deployment expenses, forecasted investment incurred monthly expenses, forecasted revenues. The user also needs to determine some total expenses per month and the remaining cash after suppliers are paid. Of course, those conclusions have no scientific foundation (this issue is one of the near future concerns) and are limited to the few researched traces. The aim was to show what kind of new knowledge can be learned from the models we produce. Also, the workflow perspective is not yet included in this view.

## 5. Evaluation of the Approach

The experimental evaluation of our approach was performed using as subjects:

- bachelor and master level students at Babes-Bolyai University of Cluj-Napoca and at the West University of Timisoara.
- expert users in Cluj-Napoca, Romania. We involved in our experiments so far a number of 7 experts (of which: 3 work in loan granting departments of different banks on various decision levels, 2 are expert accountants, 1 is working in auditing and 1 is a company manager that has a long history of loan contracting).

The experiment framework we set up requires the next steps to be followed:

- a) kick-off discussion during which the decision-aware system is introduced to the users. The moderator performs a short demonstration on how the simulation data is organized, explains the fact that each textbox is blank until selected and how the calculations can be performed within the software. He also explains to the users the decision that needs to be made (loan contracting in this particular instance) and outlines the goal (if the user needs to decide only on one value or if he needs to make a complete decision).
- b) first individual usage of the software. At this step each user logs in with the “user” username (equivalent of beginner). The focus of this first contact is to get

- a hands-on experience on how the software works. All traces produced at this stage are disregarded.
- c) second discussion with all the users. Any question regarding difficulties in using the software is answered.
  - d) second individual usage of the software. All users are required to evaluate themselves either as a beginner in the decision at hand (and needs to log in with “user” username) or as an intermediate (and needs to log in with “intermediate” username). We classify those traces as relevant for our experiment.
  - e) third discussion with all the users. The concept of PDM model is now introduced and the underlying goal of the experiment is revealed. The running example of our approach is presented and one expert trace is discussed with the participants. A random trace of the logs is selected and the PDM model is built. The participants are required to recognize if the model belongs to one of them. The expert and the user PDM models are shown in parallel and argued.
  - f) the exit questionnaire is applied. It contains seven multiple choice questions and one open question. More insights are provided in sub-section 5.2.

There are two goals that we try to reach by such an experiment. The first one is to gather a large number of traces (performed by users at various levels of knowledge), that can be compared and mined. The second one is to determine the user’s reaction to our approach and their understanding of: this new knowledge acquisition method; and of the model produced. Therefore, we are able to provide a quantitative and qualitative evaluation of our approach based on the experiments we performed so far.

### **5.1. Quantitative Evaluation**

The main focus of this sub-section is to evaluate the traces produced in the experiment and the match between the data in the logs and the models produced. As we already showed in [Vanderfeesten, 2008] the completeness issue for converting a PDM to a workflow was already demonstrated. Therefore, we only needed to evaluate two more aspects:

- a) does the software log each of the user’s actions?
- b) is the activity data, stored in the logs, completely and correctly displayed in the PDM model.

The first issue was solved by rigorously testing the decision-aware software. All the actions of the users are logged. We also performed destructive testing to check if the logging is performed when the user is trying to sabotage the experiment (this may occur if students are to be graded according to their performance in the decision scenario). For example, the software does not allow a user to add a data item to the calculation string and then immediately add another without adding a mathematical operand.

By manually checking each of the traces and comparing the automatically mined PDM with the log data we tried to determine the performance of the mining application. There were several exceptions that had to be dealt with. For example, at first we relied on the fact that in the logs a data item added to the calculation string

will be followed by an operand but that was not always the case. The mining algorithm was adjusted so it can now deal with this.

Another concern, still unanswered at this point, is the comparison of different models. This issue was already dealt with in process mining, where one can find various metrics. The existing approaches are extremely difficult to apply to decision mining due to the heterogeneity of mental processes. Process mining relies on the fact that a large number of traces in a log will be identical (or at least highly similar) while for a mental process of humans this is highly unlikely (may be the case only for very simple and structured decisions). So far we manually checked for local patterns that show up in the mined models. A more in-depth research on this aspect still needs to be conducted and will be presented in future papers.

## 5.2. Qualitative Evaluation

Qualitative evaluation was our main concern at this stage of the project. It relies on the use of a questionnaire and tries to answer several questions:

- a) is the model easily understandable?
- b) does the model depict knowledge that otherwise would be hard to get?
- c) can the user learn from such a model?

The multiple choice questions with four options (1 = very low, 2 = low, 3 = high and 4 = very high) in the questionnaire are:

- 1) How much of the PDM model introduced after the software test you can understand?
- 2) How much of the PDM model introduced after the software usage resembles your process?
- 3) Do you think that a PDM makes your knowledge explicit?
- 4) How much of your knowledge, about the loan contracting decision, would you be able to represent by yourself, using various representations (without the use of decision mining approach)?
- 5) Did the expert trace introduced as a PDM advance your knowledge on the loan contracting decision?
- 6) Did the expert trace introduced as a PDM reveal aspects of the decision you did not consider while performing the decision by yourself?

There is a seventh question with only two answer options (1 = yes and 2 = no) (“Did this experiment advance your knowledge in any way?”). It is followed by an open question which requires the user to explain how.

We are looking for high values at questions 1, 3, 5 and 6. This would show that the user understands easily the knowledge depicted as a PDM model and that he actually learns something by looking at such a model by revealing aspect he never considered before. We are looking for low values at question 4 because this would show that the users find it difficult to formalize the knowledge they possess and communicate it to others in any other way than by speaking.

The second question cannot be directly interpreted. Its purpose is to match the qualitative self assessment with the quantitative one. Basically, if we derive a high



average score at this question the mined models should show a larger number of common data items. If the average is low, then we expect a high heterogeneity of models.

The latest experiment involved a number of 33 master students in the West University of Timisoara. The results are:

*Table 2 Questionnaire results of latest experiment involving 33 intermediate decision makers*

Question no.	Q1	Q2	Q3	Q4	Q5	Q6	Q7
answer 1 (nothing)	0	0	2	0	0	1	26
answer 2 (a small part)	5	22	9	23	15	13	6
answer 3 (a large part)	26	10	22	10	14	18	
answer 4 (completely)	2	0	0	0	4	0	
no answer	0	1	0	0	0	1	1
<b>Average score</b>	<b>2,91</b>	<b>2,31</b>	<b>2,61</b>	<b>2,30</b>	<b>2,67</b>	<b>2,53</b>	<b>1,19</b>

One can easily see that the large majority of users can understand the most of a PDM model (Q1) and believe that makes most of the knowledge explicit (Q3).

The user's opinions are split almost evenly about how much they learned about loan contracting by participating in the experiment (Q5) and about how many things they did not consider in the first place might worth considering after all (Q6). The point we are making, based on the results for those questions, is that (with one exception) all the students gained, more or less, some new knowledge about the decision process of loan contracting.

The result for Q4 comes to strengthen our assertion. It reveals the fact that the largest part of the users finds it difficult to formalize their knowledge. However, this aspect requires further research because all the subjects of this experiment were second year master students in the field of Business and had no training in knowledge acquisition and representation. On the other hand, in earlier experiments conducted on second year master students in the field of Business Information Systems (which had training in knowledge acquisition) we discovered similar results, but found out that the users lacked a lot of knowledge about the business related aspects of the experiment.

## 6. Conclusions and Future Research

This paper introduces a complete framework aimed at making explicit the knowledge used in a business decision process. We log the interaction of a decision maker with a system that puts the user in the place of a company manager challenged with various simulated decisions. The log file is then mined and a Product Data Model (PDM) and eventually a workflow model can be created. The advantage of our approach is that,

when needed to investigate a large number of subjects, it is much faster, less expensive and produces more objective results than classical knowledge acquisition methods such as interviews, and questionnaires.

For making the mined knowledge explicit we use a PDM model that depicts: a) which data items, available in the simulation scenario, were considered important and relevant by the user; and b) the new data items derived based on other data. We validated our approach (software, mining tool and the models we produce) by experiments involving expert users and second year master and bachelor students. The qualitative assessment, based on the experiments we conducted, is that the PDM is easy to read and understand and that by going through an expert's model one can improve his knowledge about the decision process.

From a PDM or a workflow model produced using this approach, the people who want a better insight into a specific real life decision (like managers, students) can further their knowledge by reading and understanding a process performed by an expert. Our approach can also be used by professors for evaluating the progress in decision making training (by comparing the 'before' and 'after' models) or by professionals interested in an alternate knowledge extraction tool.

One of our major concerns, that will be investigated in the near future, is the integration of PDM models from different experts (i.e. building a model that aggregates individual traces) so that patterns present in different traces can be identified and pointed out to external observers. Due to the extremely different approaches of different users to the same decision process, standard process mining algorithms (Alpha++, Heuristic, Genetic and Fuzzy mining algorithms) output unusable spaghetti-like models. A new approach, tailored to the particularities of mental actions and processes, is required. We will also focus our future research effort on evaluating existing process model derivation algorithms and finding some criteria for selecting the fittest.

**Acknowledgments.** This work was supported by CNCSIS-UEFISCSU, project number PN II-RU TE code 292, number 52/2010.

## References

- [CW, 1998] Cook, J.E., Wolf, A.L.: Discovering Models of Software Processes from Event-Based Data. *ACM Transactions on Software Engineering and Methodology* 7(3), 215--249 (1998).
- [GW, 2004] Goodwin, P., Wright, G.: *Decision Analysis for Management Judgment*, 3rd edition. Wiley, Chichester (2004).
- [Orlicky, 1972] Orlicky, J.A.: Structuring the Bill of Materials for MRP. *J. Production and Inventory Management*, 19--42, (1972)
- [PM, 2010] Petruşel, R., Mican, D.: Mining Decision Activity Logs. In Abramowicz, W., Tolksdorf, R., Węcel, K. (eds.) *BIS 2010 Workshops*. LNBIP, vol. 57, pp. 67–79. Springer Verlag, (2010)
- [Power, 2004] Power, D. J., "Decision Support Systems: From the Past to the Future," *Proceedings of the 2004 Americas Conference on Information Systems*, New York, NY, August 6-8, 2004a, 2025-2031, (2004).

- [RLvdA, 2003] Reijers, H.A., Limam Mansar, S., van der Aalst, W.M.P.: Product-Based Workflow Design. *Journal of Management Information Systems*. 20, 229–262 (2003)
- [RvdA, 2006] Rozinat, A., van der Aalst, W.M.P.: Decision Mining in ProM. In: Dustdar, S., Faideiro, J.L., Sheth, A. (eds) *BPM 2006*. LNCS vol. 4102, pp. 420–425. Springer, Berlin (2006).
- [Simon, 1960] Simon, H.A.: *The New Science of Management Decision*. Harper and Row, New York (1960)
- [TSD, 2010] Turban E., Sharda R., Delen D.: *Decision Support and Business Intelligence Systems*, 9th Edition, Prentice Hall, New Jersey, (2010).
- [Vanderfeesten, 2008] Vanderfeesten, I.: *Product-Based Design and Support of Workflow Processes*. Eindhoven University of Technology, Eindhoven (2009)
- [vdAvH, 2002] van der Aalst, W.M.P., van Hee, K.: *Workflow Management: Models, Methods and Systems*. MIT Press, Cambridge (2002).
- [vdAvDHMSW, 2003] van der Aalst, W.M.P., van Dongen, B.F., Herbst, J., Maruster, L., Schimm, G., Weijters, A.J.M.M.: *Workflow Mining: A Survey of Issues and Approaches*. *J. Data and Knowledge Engineering*. 47, 237–267 (2003)
- [vdAW, 2004] van der Aalst, W.M.P., Weijters, A.J.M.M.: *Process Mining*. *Computers in Industry*. 53(3), 231–244 (2004).

Working Papers Beta 2009 - 2011

nr.	Year	Title	Author(s)
342	2011	<a href="#">Optimal Inventory Policies with Non-stationary Supply Disruptions and Advance Supply Information</a>	Bilge Atasoy, Refik Güllü, Tarkan Tan
341	2011	<a href="#">Redundancy Optimization for Critical Components in High-Availability Capital Goods</a>	Kurtulus Baris Öner, Alan Scheller-Wolf Geert-Jan van Houtum
340	2011	<a href="#">Making Decision Process Knowledge Explicit Using the Product Data Model</a>	Razvan Petrusel, Irene Vanderfeesten, Cristina Claudia Dolean, Daniel Mican
339	2010	<a href="#">Analysis of a two-echelon inventory system with two supply modes</a>	Joachim Arts, Gudrun Kiesmüller
338	2010	<a href="#">Analysis of the dial-a-ride problem of Hunsaker and Savelsbergh</a>	Murat Firat, Gerhard J. Woeginger
335	2010	<a href="#">Attaining stability in multi-skill workforce scheduling</a>	Murat Firat, Cor Hurkens
334	2010	<a href="#">Flexible Heuristics Miner (FHM)</a>	A.J.M.M. Weijters, J.T.S. Ribeiro
333	2010	<a href="#">An exact approach for relating recovering surgical patient workload to the master surgical schedule</a>	P.T. Vanberkel, R.J. Boucherie, E.W. Hans, J.L. Hurink, W.A.M. van Lent, W.H. van Harten
332	2010	<a href="#">Efficiency evaluation for pooling resources in health care</a>	Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Nelly Litvak
331	2010	<a href="#">The Effect of Workload Constraints in Mathematical Programming Models for Production Planning</a>	M.M. Jansen, A.G. de Kok, I.J.B.F. Adan
330	2010	<a href="#">Using pipeline information in a multi-echelon spare parts inventory system</a>	Christian Howard, Ingrid Reijnen, Johan Marklund, Tarkan Tan
329	2010	<a href="#">Reducing costs of repairable spare parts supply systems via dynamic scheduling</a>	H.G.H. Tiemessen, G.J. van Houtum
328	2010	<a href="#">Identification of Employment Concentration and Specialization Areas: Theory and Application</a>	F.P. van den Heuvel, P.W. de Langen, K.H. van Donselaar, J.C. Fransoo
327	2010	<a href="#">A combinatorial approach to multi-skill workforce scheduling</a>	Murat Firat, Cor Hurkens

326	2010	<a href="#">Stability in multi-skill workforce scheduling</a>	Murat Firat, Cor Hurkens, Alexandre Laugier
325	2010	<a href="#">Maintenance spare parts planning and control: A framework for control and agenda for future research</a>	M.A. Driessen, J.J. Arts, G.J. v. Houtum, W.D. Rustenburg, B. Huisman
324	2010	<a href="#">Near-optimal heuristics to set base stock levels in a two-echelon distribution network</a>	R.J.I. Basten, G.J. van Houtum
323	2010	<a href="#">Inventory reduction in spare part networks by selective throughput time reduction</a>	M.C. van der Heijden, E.M. Alvarez, J.M.J. Schutten
322	2010	<a href="#">The selective use of emergency shipments for service-contract differentiation</a>	E.M. Alvarez, M.C. van der Heijden, W.H. Zijm
321	2010	<a href="#">Heuristics for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering in the Central Warehouse</a>	B. Walrave, K. v. Oorschot, A.G.L. Romme
320	2010	<a href="#">Preventing or escaping the suppression mechanism: intervention conditions</a>	Nico Dellaert, Jully Jeunet.
319	2010	<a href="#">Hospital admission planning to optimize major resources utilization under uncertainty</a>	R. Seguel, R. Eshuis, P. Grefen.
318	2010	<a href="#">Minimal Protocol Adaptors for Interacting Services</a>	Tom Van Woensel, Marshall L. Fisher, Jan C. Fransoo.
317	2010	<a href="#">Teaching Retail Operations in Business and Engineering Schools</a>	Lydie P.M. Smets, Geert-Jan van Houtum, Fred Langerak.
316	2010	<a href="#">Design for Availability: Creating Value for Manufacturers and Customers</a>	Pieter van Gorp, Rik Eshuis.
315	2010	<a href="#">Transforming Process Models: executable rewrite rules versus a formalized Java program</a>	Bob Walrave, Kim E. van Oorschot, A. Georges L. Romme
314	2010	<a href="#">Getting trapped in the suppression of exploration: A simulation model</a>	S. Dabia, T. van Woensel, A.G. de Kok
313	2010	<a href="#">A Dynamic Programming Approach to Multi-Objective Time-Dependent Capacitated Single Vehicle Routing Problems with Time Windows</a>	

2010		
312	2010	<a href="#">Tales of a So(u)rcerer: Optimal Sourcing Decisions Under Alternative Capacitated Suppliers and General Cost Structures</a>
		Osman Alp, Tarkan Tan
311	2010	<a href="#">In-store replenishment procedures for perishable inventory in a retail environment with handling costs and storage constraints</a>
		R.A.C.M. Broekmeulen, C.H.M. Bakx
310	2010	<a href="#">The state of the art of innovation-driven business models in the financial services industry</a>
		E. Lüftenegger, S. Angelov, E. van der Linden, P. Grefen
309	2010	<a href="#">Design of Complex Architectures Using a Three Dimension Approach: the CrossWork Case</a>
		R. Seguel, P. Grefen, R. Eshuis
308	2010	<a href="#">Effect of carbon emission regulations on transport mode selection in supply chains</a>
		K.M.R. Hoen, T. Tan, J.C. Fransoo, G.J. van Houtum
307	2010	<a href="#">Interaction between intelligent agent strategies for real-time transportation planning</a>
		Martijn Mes, Matthieu van der Heijden, Peter Schuur
306	2010	<a href="#">Internal Slackening Scoring Methods</a>
		Marco Slikker, Peter Borm, René van den Brink
305	2010	<a href="#">Vehicle Routing with Traffic Congestion and Drivers' Driving and Working Rules</a>
		A.L. Kok, E.W. Hans, J.M.J. Schutten, W.H.M. Zijm
304	2010	<a href="#">Practical extensions to the level of repair analysis</a>
		R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
303	2010	<a href="#">Ocean Container Transport: An Underestimated and Critical Link in Global Supply Chain Performance</a>
		Jan C. Fransoo, Chung-Yee Lee
302	2010	<a href="#">Capacity reservation and utilization for a manufacturer with uncertain capacity and demand</a>
		Y. Boulaksil; J.C. Fransoo; T. Tan
300	2009	<a href="#">Spare parts inventory pooling games</a>
		F.J.P. Karsten; M. Slikker; G.J. van Houtum
299	2009	<a href="#">Capacity flexibility allocation in an outsourced supply chain with reservation</a>
		Y. Boulaksil, M. Grunow, J.C. Fransoo
298	2010	<a href="#">An optimal approach for the joint problem of level of repair analysis and spare parts stocking</a>
		R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
297	2009	<a href="#">Responding to the Lehman Wave: Sales Forecasting and Supply Management during the Credit Crisis</a>
		Robert Peels, Maximiliano Udenio, Jan C. Fransoo, Marcel Wolfs, Tom Hendrikx
296	2009	<a href="#">An exact approach for relating recovering surgical patient workload to the master surgical schedule</a>
		Peter T. Vanberkel, Richard J. Boucherie, Erwin W. Hans, Johann L. Hurink, Wineke A.M. van Lent, Wim H. van Harten
295	2009	<a href="#">An iterative method for the simultaneous optimization of repair decisions and spare parts stocks</a>
		R.J.I. Basten, M.C. van der Heijden, J.M.J. Schutten
294	2009	<a href="#">Fujaba hits the Wall(-e)</a>
		Pieter van Gorp, Ruben Jubeh, Bernhard Grusie, Anne Keller

293	2009	<a href="#">Implementation of a Healthcare Process in Four Different Workflow Systems</a>	R.S. Mans, W.M.P. van der Aalst, N.C. Russell, P.J.M. Bakker
292	2009	<a href="#">Business Process Model Repositories - Framework and Survey</a>	Zhiqiang Yan, Remco Dijkman, Paul Grefen
291	2009	<a href="#">Efficient Optimization of the Dual-Index Policy Using Markov Chains</a>	Joachim Arts, Marcel van Vuuren, Gudrun Kiesmuller
290	2009	<a href="#">Hierarchical Knowledge-Gradient for Sequential Sampling</a>	Martijn R.K. Mes; Warren B. Powell; Peter I. Frazier
289	2009	<a href="#">Analyzing combined vehicle routing and break scheduling from a distributed decision making perspective</a>	C.M. Meyer; A.L. Kok; H. Kopfer; J.M.J. Schutten
288	2009	<a href="#">Anticipation of lead time performance in Supply Chain Operations Planning</a>	Michiel Jansen; Ton G. de Kok; Jan C. Fransoo
287	2009	<a href="#">Inventory Models with Lateral Transshipments: A Review</a>	Colin Paterson; Gudrun Kiesmuller; Ruud Teunter; Kevin Glazebrook
286	2009	<a href="#">Efficiency evaluation for pooling resources in health care</a>	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
285	2009	<a href="#">A Survey of Health Care Models that Encompass Multiple Departments</a>	P.T. Vanberkel; R.J. Boucherie; E.W. Hans; J.L. Hurink; N. Litvak
284	2009	<a href="#">Supporting Process Control in Business Collaborations</a>	S. Angelov; K. Vidyasankar; J. Vonk; P. Grefen
283	2009	<a href="#">Inventory Control with Partial Batch Ordering</a>	O. Alp; W.T. Huh; T. Tan
282	2009	<a href="#">Translating Safe Petri Nets to Statecharts in a Structure-Preserving Way</a>	R. Eshuis
281	2009	<a href="#">The link between product data model and process model</a>	J.J.C.L. Vogelaar; H.A. Reijers
280	2009	<a href="#">Inventory planning for spare parts networks with delivery time requirements</a>	I.C. Reijnen; T. Tan; G.J. van Houtum
279	2009	<a href="#">Co-Evolution of Demand and Supply under Competition</a>	B. Vermeulen; A.G. de Kok
278	2010	Toward Meso-level Product-Market Network Indices for Strategic Product Selection and (Re)Design Guidelines over the Product Life-Cycle	B. Vermeulen, A.G. de Kok
277	2009	<a href="#">An Efficient Method to Construct Minimal Protocol Adaptors</a>	R. Seguel, R. Eshuis, P. Grefen
276	2009	<a href="#">Coordinating Supply Chains: a Bilevel Programming Approach</a>	Ton G. de Kok, Gabriella Muratore
275	2009	<a href="#">Inventory redistribution for fashion products under demand parameter update</a>	G.P. Kiesmuller, S. Minner
274	2009	<a href="#">Comparing Markov chains: Combining aggregation and precedence relations applied to sets of states</a>	A. Basic, I.M.H. Vliegen, A. Scheller-Wolf
273	2009	<a href="#">Separate tools or tool kits: an exploratory study of engineers' preferences</a>	I.M.H. Vliegen, P.A.M. Kleingeld, G.J. van Houtum

272	2009	<a href="#">An Exact Solution Procedure for Multi-Item Two-Echelon Spare Parts Inventory Control Problem with Batch Ordering</a>	Engin Topan, Z. Pelin Bayindir, Tarkan Tan
271	2009	<a href="#">Distributed Decision Making in Combined Vehicle Routing and Break Scheduling</a>	C.M. Meyer, H. Kopfer, A.L. Kok, M. Schutten
270	2009	<a href="#">Dynamic Programming Algorithm for the Vehicle Routing Problem with Time Windows and EC Social Legislation</a>	A.L. Kok, C.M. Meyer, H. Kopfer, J.M.J. Schutten
269	2009	<a href="#">Similarity of Business Process Models: Metrics and Evaluation</a>	Remco Dijkman, Marlon Dumas, Boudewijn van Dongen, Reina Kaarik, Jan Mendling
267	2009	<a href="#">Vehicle routing under time-dependent travel times: the impact of congestion avoidance</a>	A.L. Kok, E.W. Hans, J.M.J. Schutten
266	2009	<a href="#">Restricted dynamic programming: a flexible framework for solving realistic VRPs</a>	J. Gromicho; J.J. van Hoorn; A.L. Kok; J.M.J. Schutten;

Working Papers published before 2009 see: <http://beta.ieis.tue.nl>