Implementing a Decision-Aware System for Loan Contracting Decision Process

Răzvan PETRUŞEL, Daniel MICAN, Cristina Claudia DOLEAN
Business Information Systems Department,
Faculty of Economics and Business Administration.
Babeş-Bolyai University, Cluj-Napoca, Romania
{razvan.petrusel, daniel.mican, cristina.dolean}@econ.ubbcluj.ro

The paper introduces our work related to the design and implementation of a decision-aware system focused on the loan contracting decision process. A decision-aware system is a software that enables the user to make a decision in a simulated environment and logs all the actions of the decision maker while interacting with the software. By using a mining algorithm on the logs, it creates a model of the decision process and presents it to the user. The main design issue introduced in the paper is the possibility to log the mental actions of the user. The main implementation issues are: user activity logging programming and technologies used. The first section of the paper introduces the state-of-the-art research in process mining and the framework of our research; the second section argues the design of the system; the third section introduces the actual implementation and the fourth section shows a running example.
Keywords: *Decision-Aware Systems, Decision Activity Logs, Decision Mining, Codeigniter, JSON*

1 Introduction

Decision making is a process that consists of a sequence of actions (mostly mental) performed by a decision maker. If a manager is provided with software containing all the necessary data for a certain decision, the user activity within the software is a physical instance of the reasoning patterns. The actions within such software should be logged, thus capturing the knowledge employed by the user. If the system is capable of providing a feedback decision process reasoning model, based on the logged system interaction, the software becomes decision-aware. Decision mining is the activity that, based on logs of user interaction with decision-aware software, extracts and creates a model of the decision making process that shows the mental activity sequence (decision workflow) performed by the manager during the process.

We seek to enable capturing (in logs) and communications (through models) of thinking and reasoning patterns of individual decision makers. Based on the experimental validation, another output of the project will be a set of reference models for all the major decisions in an enterprise.

The target audience of our approach is: professionals in knowledge acquisition and researchers in economics. The first category can benefit from this novel knowledge acquisition method. The major benefit is that we provide a quicker, sometimes more accurate and less expensive (compared to the classic methods like questionnaires, interviews, etc.) instrument for extracting and depicting knowledge from large samples of individuals. Also, this method allows a researcher to cover decision making from the process perspective. So far the process perspective of decision making is limited to a theoretical approach and less, as far as we are aware of, to creating process models for particular decisions in an enterprise.

Our research draws on the previous work in two major fields: simulation (of workflows and overall enterprise business) and process mining.

The simulation of workflows aims to improve the business processes performed within an enterprise. Some approaches are introduced in [1] [2] [3] [4] [5]. The goal of such a simulation is to find a better way to use the resources of the enterprise, to redistribute the tasks within the enterprise, etc. The decision elements are actually the forks (splits) in the model where one of the paths in the model needs to be chosen. The goal of such a simulation is to optimize the decision nodes in order to improve the throughput time of a case.

Process mining is an activity focused on finding nontrivial and novel knowledge and distilling a structured process model from the low level logs of transactional systems [6]. The aim of process mining is to build a model of the workflow of physical activities performed by the employees and systems of the enterprise. The first research in finding an algorithm that will convert a log into a workflow model started in 1990 [7], [8]. Currently the most cited work belongs to Van der Aalst and the books that set the foundation for the state-of-the-art are [9], [10].

The virtual environment will cover all the major departments of an enterprise: purchasing, production, sales, human resources, investments, and financial management [11]. The departments that will require the user to make decisions are introduced in Figure 1.

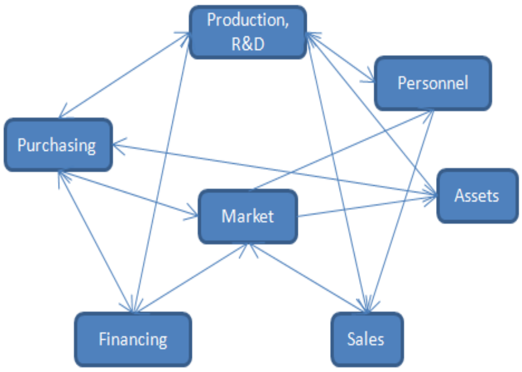


Fig. 1. Departments of the virtual enterprise requiring decisions

However, the user must concentrate on making only one decision at a time. Therefore, the general design of the virtual environment must allow the clear separation between decisions while allowing the usage of data and decisions in one department for all other departments. The architecture of the decision-aware system is introduced in Figure 2.

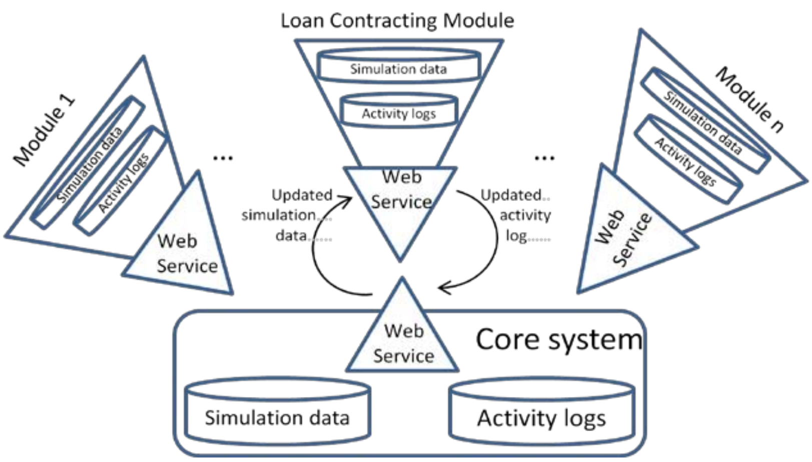


Fig. 2. Architecture of the enterprise-wide decision-aware system

It can be seen in Figure 2 that there needs to be a separate module for each decision process. The example shown in the next section of this paper focuses on the implementation of the Loan Contracting Module. This specialization on many decision modules gives us the opportunity to clearly assign all the activities of the decision maker to only one decision process. The design on the enterprise-wide system will require us to develop specialized simulations for each decision process we will research. Each such module will use web services to transfer decision activity logs to the core system and to receive data updates for the simulation data. The activity logs will be transferred to the central log database on the core system after each session. If the user makes a decision in one module, the scenario data in all the other connected modules might require an update (for example, if the user decides to employ more people in one module the scenario data in production module will be update according the new employee number).

Each individual module can be created using a different approach. The only restriction is that each module needs to output the activity logs in the standard format used in the core system. The framework architecture of an individual decision module is shown in Figure 3. Basically, the decision module will need to have a user interface that the decision maker will use in order to interact with the system. There also should be a security component that will allow the user access control. The application logic consists mainly of model management, scenario data and data deriving components. The logging of the user interaction will be done by the dedicated component.

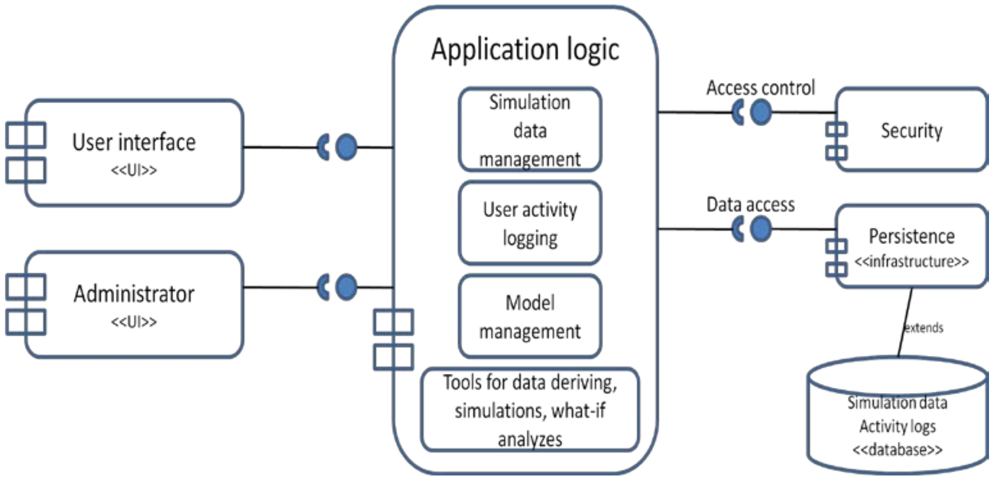


Fig. 3. Architecture of a decision module of the decision-aware system

The general approach over the lifecycle of a decision-aware system is introduced in Figure 4.

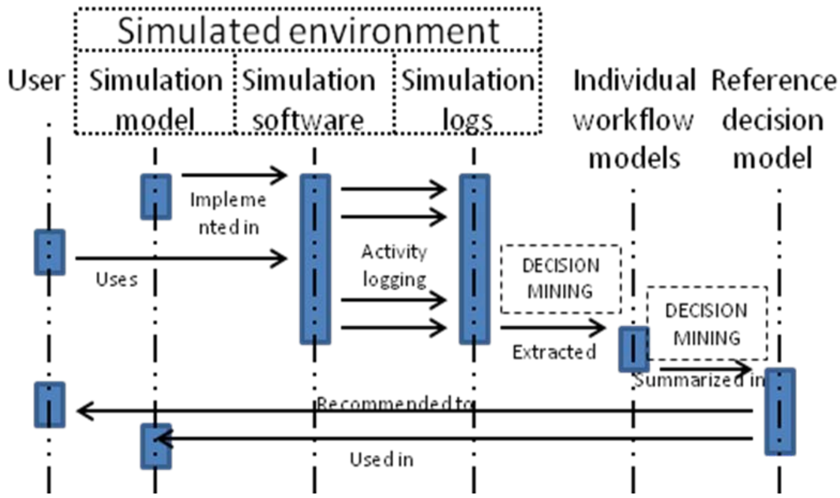


Fig. 4. Lifecycle of a decision-aware system

The simulated environment has three components. The simulation model comprises the knowledge extracted from the experts regarding the decision that will be simulated within the module. This consists of the data that is required for the decision, the rules and restriction that apply and a general workflow that can be used. The simulation software is the actual system that must provide all the data to the decision maker (ranging from unimportant to critical) and must enable the logging of user activities within the software. The simulation logs are tables containing all the user actions. Each module simulates only one decision circumstance; therefore the logs produced will be associated with only one decision process and only one decision type. There are two outputs of the decision mining process: individual decision process models (one model for each trace in the logs) and a reference decision model (one model for all the traces in the logs). There should also be a tool and some metrics that will allow the individual decision process model to be compared with the reference model. The reference model will be used in the next software iteration to update and improve the simulation model.
The architecture of the loan contracting decision module presented in the next section is introduced in Figure 6. It is one of the modules of the enterprise-wide system presented in Figure 2 and follows the framework architecture presented in Figure 3. While developing this module we used the lifecycle introduced in Figure 4.

2 The loan contracting decision-aware system implementation

Some issues on key points of the proposed approach are:
- in order to build an accurate simulation, it is essential to thoroughly explore the decision problem and context with the help of experts. This primarily involves precise determination of all data and information that should be presented to the user by the software (internal and external to the company).
- an important issue affecting the quality of logs is the data presentation in the user interface. Data can be presented either centralised (on only one page), either decentralised (divided over several pages with links between them). The advantage of centralised presentation is that users no longer have to navigate within the application, thereby producing a log that contains only the actual action. The disadvantage of this approach is that pages will have a large number of elements that confuse the user. If data presentation is decentralised, the user will not be able to view all data in the same pages and numerous intermediate steps of navigation will be required. This can impede on data comparisons for example.
- logging user activity is critically influenced by the tools employed. If using low-level tools (logging the controls clicked by the user) the data format is tabular, easily usable in mining algorithms. If using high-level tools (e.g. eye tracking) the output is a "heat map" that contains the locations where the user's gaze concentrated (symbolized by the colours of different intensities) and lines representing the eye’s movements (indicating user attention moving from one element to the next one).
- the main problem of mining decision logs is the mathematical foundation and the model’s representation format. We considered that the most appropriate format is coloured Petri nets (CPN). These networks are built based on solid mathematical foundations and have demonstrated the ability to represent the workflows in research related to process mining.
- in the model generated after the mining, the biggest challenge is to balance a too general or too detailed presentation of the activities.
Some of the assumptions used in setting up the project:
- The system should be similar to a Decision Support System. It should provide the user with all the critical data ranging from trivial to critical and should give the user the possibility to derive information and new data based on the simulation data;
- To perform an effective decision mining it is essential to "force" the user to mentally divide the decision process into basic mental

activities. To build such software, we have to reconsider the entire software development process because we add a new key objective that has to be achieved by the system.
- The software has to track and log all data used by the user. If new data is derived, the system should log which are the source data and the formula used.
- The software should be constructed so that all the mental activities of the decision-maker will take place within the application. The user needs no data other than those available in the system and should not be allowed access to other sources of information.

Decision mining is based on several assumptions:
- Every action in the system log refers to a single mental activity and is a well defined step in decision-making process.
- Each logged action refers to one and only one well-defined decision process trace. A full decision process is considered one that ends by the user's choice of an alternative. Any other end transforms the process in an incomplete trace.
- Each decision-making process is performed by known decision makers;
- In case of collaborative decision-making processes, each action can be attributed to a single decision-maker and each interaction between participants can be reduced to atomic actions performed by a single person.
- Each action has an attached timestamp and actions are performed in a logical order by the decision maker.

The overall approach used in decision mining is introduced in Figure 5:

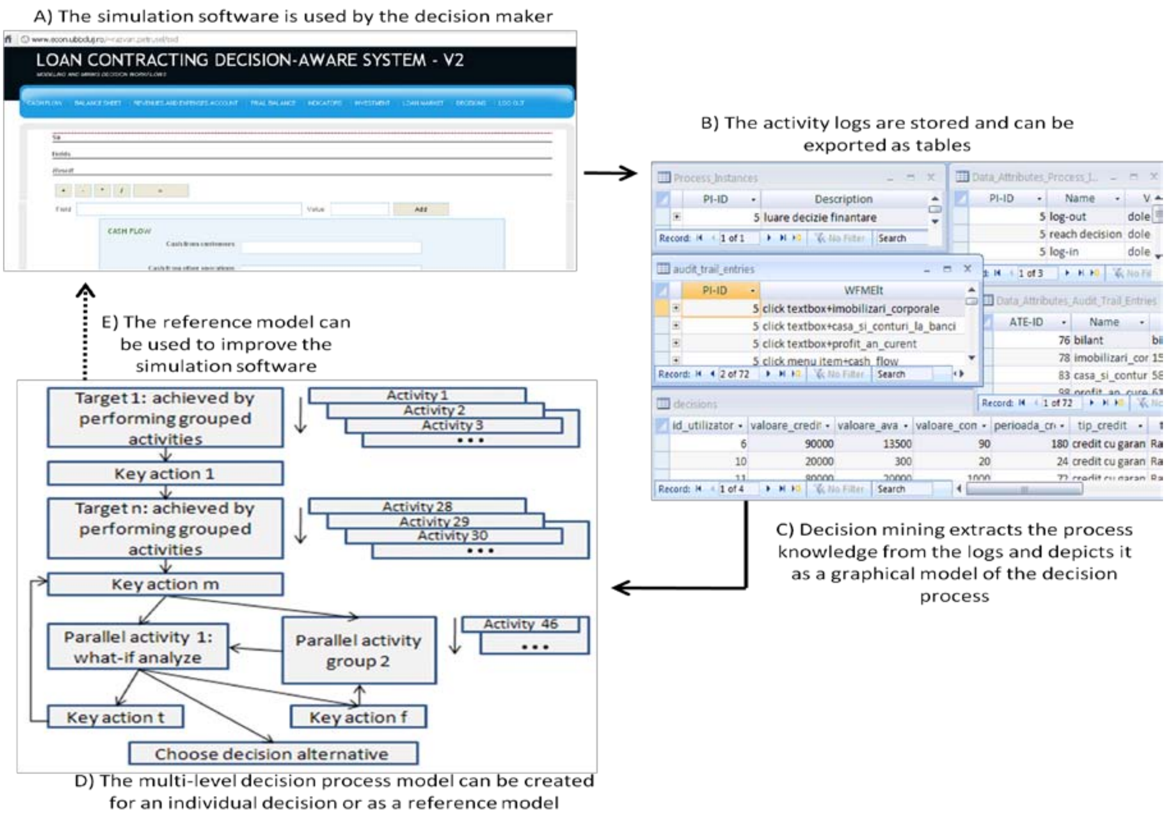


Fig. 5. Overall approach of decision mining

To implement the system we used the application framework CodeIgniter (available: http://www.codeigniter.com/). CodeIgniter is a powerful PHP framework containing a set of simple and easy to use tools for building complex web applications. This is an open source framework and aims to allow developers to develop projects more quickly than if they would start writing code from scratch. Therefore, the construction of complex and dynamic web projects is allowed because the interface provides a simple and logical struc-

ture to access a rich set of libraries for the most commonly used functions and methods. CodeIgniter is largely based on the popular Model-View-Controller architectural pattern.
Model-View-Controller (MVC) [12] is a software architecture commonly used in software engineering, that isolates the application logic from user interface. The Model component is used to represent knowledge in the form of objects or object structures. This information is used to manage information, and when changes occur to notify the associated views so they can update their content. The View component is a visual representation of the model in an appropriate form, used as an interface with users. The view is composed of all objects that the user can interact with, such as buttons, links, textbox's, etc. The Controller component makes the connection between user and system, between model and view components. Therefore, a controller receives input data from and initiates calls to the model objects while making possible the transmission of messages to the view.
Prototype (available: http://www.prototypejs.org/) is a JavaScript Framework, which aims to facilitate the development of dynamic web applications. It offers many features for developing applications with JavaScript and XMLHttpRequest in particular. It also features an Ajax toolkit and library for developing object oriented applications.
In Fig. 6 we show a description class diagram based on MVC architecture. In Class Data/Model we defined several features that we describe below. Thus, we have the GetData() function that takes the data from the specified table as a parameter and puts it into an array. GetCredit() function gets the loan data with the specified type as a parameter. GetTime() function returns the time in milliseconds and GetDobanda() function retrieves the credit data type and percentage interest.

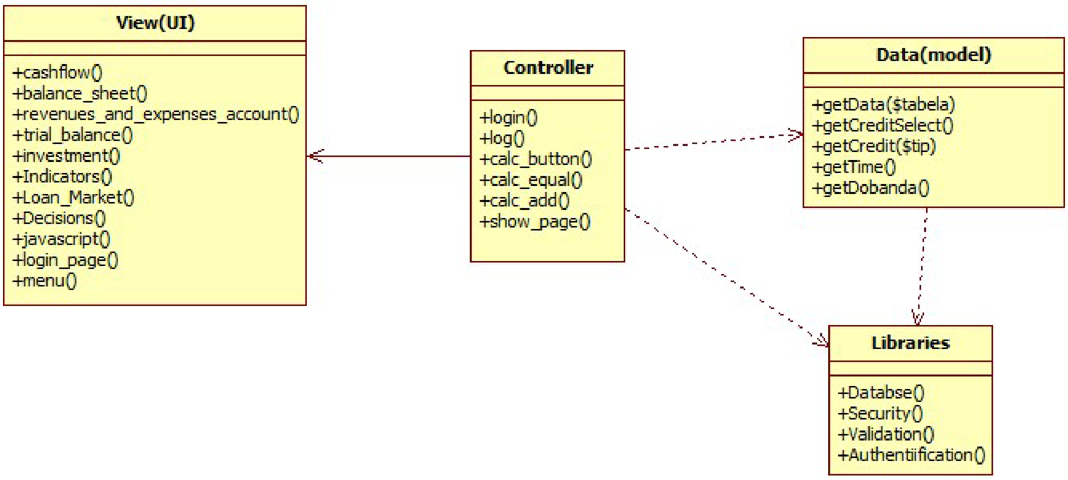


Fig. 6. Class diagram of the Loan Contracting Decision Module

In Libraries Class there are multiple database, security, validation and authentication libraries. Database is the library which creates the connection with the mysql server. Security library is used to secure data so that it does not allow attacks like SQL injection, XSS, CSRF/XSRF, etc.. Validation library contains classes for validating incoming data as input. Authentication library allows the authentication of users, checks if the username and password are correct and sets the session variable which determines whether the user is logged or not.
In the Controller class:
- login() method displays the login page,
- log() method contains the procedure to store/log in the database the events and the values in the clicked textboxes,
- calc_button() method adds the +, -, *, /, and = signs in the array used for the representation of the expression string,
- calc_add() method adds a numerical value

to in the array used for the representation of the expression string,
- calc_equal() method calculates and displays the value of expression;
- show_page()method displays the received page as a parameter in the URL.

View class contains the methods: cashflow(), balance_sheet(), revenues_and_expenses_account(), trial_balance(), investment(), indicators(), loan_market(), Decisions(), login_page(), and menu(). Each method dynamically generates a HTML page with the same name.

To log the clicks performed by the user within the software and the textbox's existing values, we retrieve all events of type “click” on each page, we filter only those that are of INPUT type and contain “text” attribute. For each click a POST method is performed through Ajax calls to a procedure that logs the events with id and value parameters.

As we have shown before, logging only the user activity is not enough. Therefore, we need to also log the values that are derived by the user during the decision making process. In order to calculate different data based on the data provided in the simulation scenario, we implemented a calculator. It can get values from textboxes in the forms of all pages in the software. When a user clicks on a textbox the system will display both the related value and an “Add” button that allows him to add the current value in the calculation string (see Figure 7). Therefore, we defined a function that is invoked by the on_click event of each button. It triggers an Ajax POST type call with parameters id and value. The result is received by the calculator in JSON format. If an error occurs the error message is displayed (error), otherwise the label “string” of the calculator is updated. The calculator data string is stored in a variable in user session and is initialized and reset at each login. With every item added to the computer, the string is interpreted, encoded in JSON format and transmitted to the client to be interpreted JavaScript.

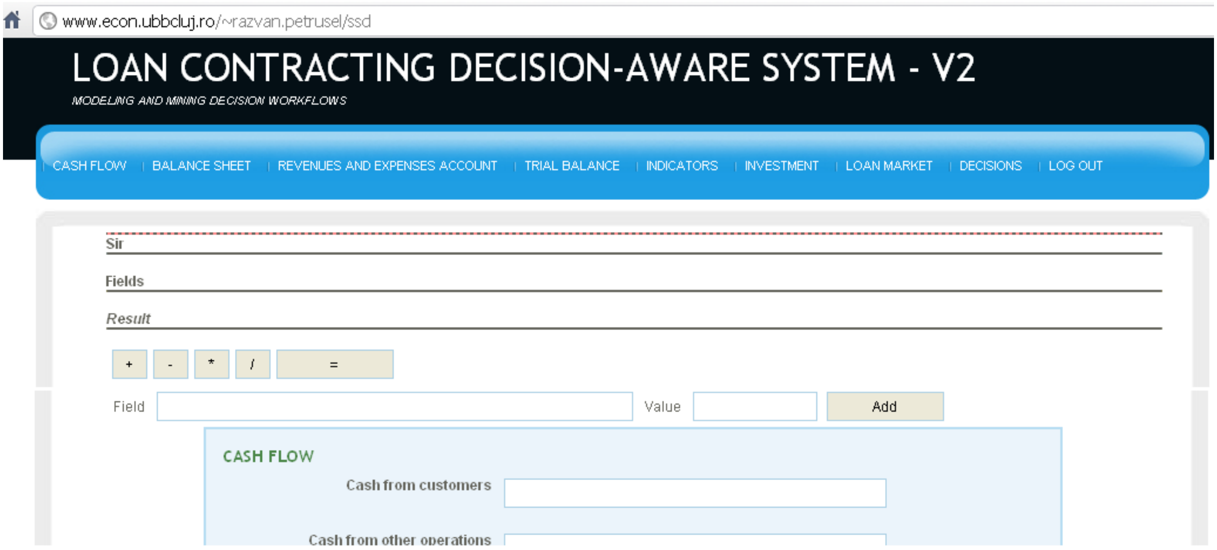


Fig. 7. User interface of the Loan Contracting Decision Module

JSON (JavaScript Object Notation) is a format for data transfer. It is easy to read/write by humans and analyzed/generated by computers. JSON is a text format that is independent of language and uses the universal conventions and data structures that are familiar to programmers acquainted with the language family C, C + +, C #, Java, JavaScript, Perl, Python, etc. Due to these properties, JSON is ideal for transferring data and is supported by all modern programming languages.

The mining activity is based on the activity logs saved in the SQL database of the decision-aware system. Those tables are exported as Microsoft Excel files and then imported in Access as tables. Five tables are used to store logs, four of which are consistent with the structure required for importing data into tables ProM Framework (available: http://prom.win.tue.nl/research/wiki/prom/start) and are introduced in Figure 8. By using the ProM Import Tool (available: http://prom.win.tue.nl/research/wiki/promimport/start) those tables can be importen into the ProM Framework and therefore all the process mining algorithms can be used for converting the logs into models.

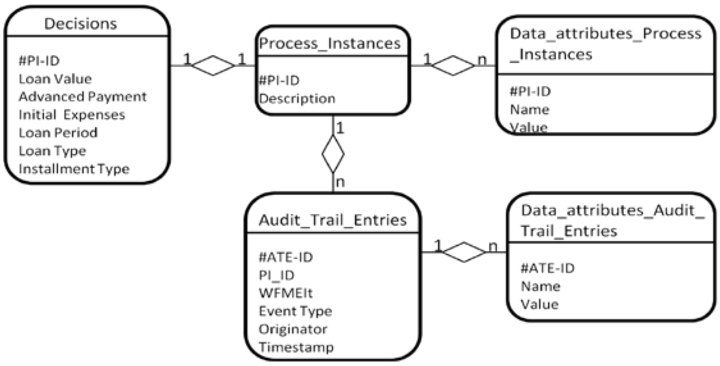


Fig. 8. Tables used for storing decision activities logs

Each record stores each user's interaction with the system (e.g. clicking on a menu or text box to view certain data, etc.). Process_Instances table contains a unique id for each process and the process name. Process name field stores the decision type in which the decision maker is involved. Since the current software only simulates the loan contracting decision, all instances of the log will show it. As other modules, simulating different decisions, are implemented this field will allow us to classify the instances for each decision. Due to the virtual environment architecture (a web service core and modules for each department of the company) only a limited number of decisions are made by accessing a module. A complete trace of a decision process must contain three actions: log-in, log out and reach decision, actions stored in the Data_Attributes_Process_Instances table. The table Audit_Trail_Entries stores the actual action performed by the user, stored in WFMEIt field as action_performed + object_type + name_of_used_control (eg. click + textbox + investment_value). The table Data_Attributes_ATE stores the values displayed by the used control (eg. the value stored in the text box valoare_investitie is 100000 lei).

The records from logs actions can be directly connected to each decision maker. The log entries are completely ordered according to the timestamp between the log-in and log-out limits of a trace. A process is considered complete if the action reach decision is stored for the trace.

Based on the activity logs, the activities are grouped into clusters (activity B in Figure 9). This activity uses domain-specific knowledge that was previously incorporated in the simulation model and methods derived from data mining to cluster the log activities. The aim of this activity is the recognition of local patterns of mental activities. This is based on rules extracted from experts in the construction phase of the simulation model. Each rule expresses conditionality between the data items presented in the simulation. The purpose of this local pattern recognition is to reveal knowledge limitations and relationships between various elements of the enterprise. After this step, all user-generated activities will be assigned to a certain number of clusters.

Activity C tries to determine relations between clusters. Basically, we try to identify the existence of preconditions (if a specific set of activities must be performed before another) or parallelism (if a number of activities belonging to several clusters may execute in random order).

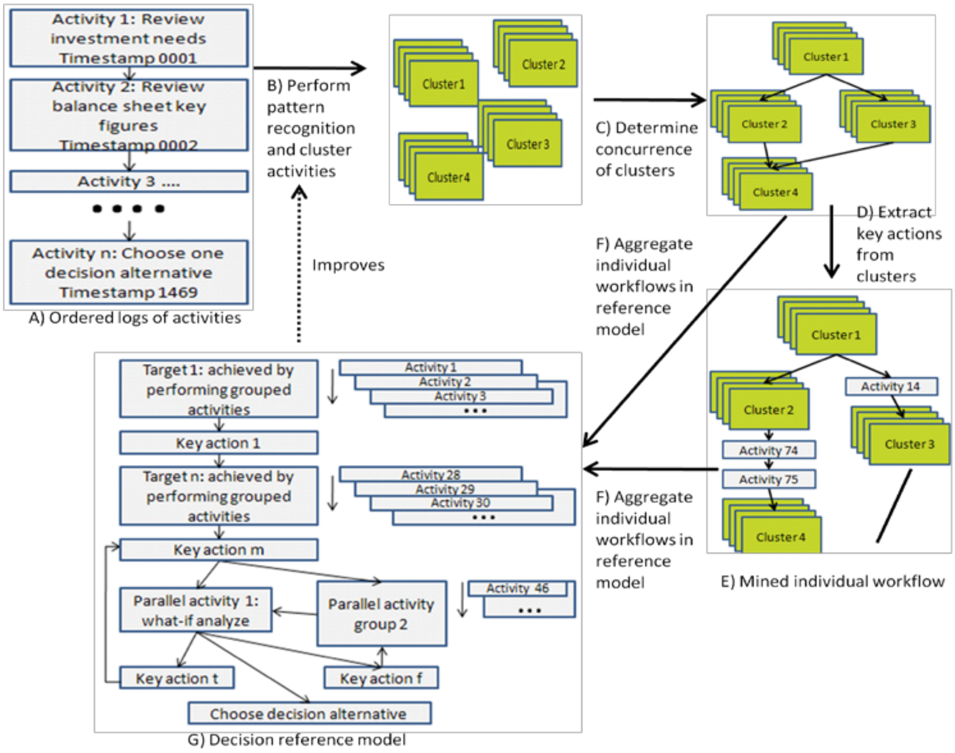


Fig. 9. Proposed approach of decision mining

Activity D checks whether there are some key actions that are mandatory for a given decision. Our current approach is to identify these activities statistically (e.g. if 95% of the 1000 users have checked a certain value in the simulation, it means that the value is critical to the decision-making process and must be depicted individually).

One of the goals of our research is to mine an individual decision-making process model. It should contain clusters of activities strictly determined based on the actions in an individual log trace. Such a model shows clusters of activities and those key actions (determined based on the logs of all users) that are present in the trace generated by the individual user.

During F activities we seek to aggregate individual models in a reference model. Aggregation is done by both by using all the traces in the logs grouped in clusters and by using the individual models mined for each user.

The decision-making process model (G) is multi-level. The overview shows the clusters of activities and the key activities. At this level the process model should only show to the user the general approach over a decision (e.g. in the loan contracting decision it can state that first the user needs to document the investment that generates the financing need, then investigate the available loans on the market, then state that the user should always check the interest rate of the loans, then review the financial position of the company. Each cluster can be further expanded into sub-clusters on several levels. The lowest level is reached when each sub-cluster is composed only of atomic activities. At the top level the problem of parallelism and the order in which the clusters are depicted is present (e.g. what-if analysis for a number of actions will be repeated several times, requiring the introduction of cyclic structures in the model).

Another approach currently under investigation is data centric.

3 The experiments and a running example

The experiments were performed by involving the final year students at bachelor and master level, resulting in two samples representing the knowledge of students near graduation of the first and second study cycles (our experiments were conducted during the last semester of courses). The first sample consisted of a number of 35 users and the second one of 12 users (according to the approximate proportions of the number of students enrolled at our university in the first and the second cycle of studies). We also used a number of four experts who have helped us build the simulation model. Therefore, the sample logs consist of 4 expert traces, 12 intermediate traces and 35 beginner traces.

We set up three usage sessions during the semester, according to the level of knowledge held by students. The first session was held at the beginning of the semester. The users had to employ only the knowledge they previously held (the level of knowledge is considered beginner) with a minimal training about the system. The second session was conducted at the middle of the semester, after the students had some training in decision-making process and strategies. We did not directly discuss the loan contracting decision or the scenario implemented in the decision-aware system (level of knowledge is considered intermediate). The third session was held at the end of the semester, after the loan contracting decision was discussed during the classes, but without mentioning specific data. In this case, the level of knowledge is considered to be advanced.

The decision scenario implemented in the decision-aware system requires the user to be the manager of an enterprise. The user is required to decide over the possibility of financing an investment by contracting a loan. The user must choose a decision alternative for: loan amount, loan period, the down payment, loan type and type of reimbursement instalments. The values inputted by each decision maker are stored in the Decisions table. All these values need to be filled-in in order to save the decision (Decision click the Save button / Log-off). If all the values are not filled-in and / or the user wants to exit without making a final decision, the log-out option is available in the menu bar. A process is considered complete if it is ended by saving all the required values for the decision variables.

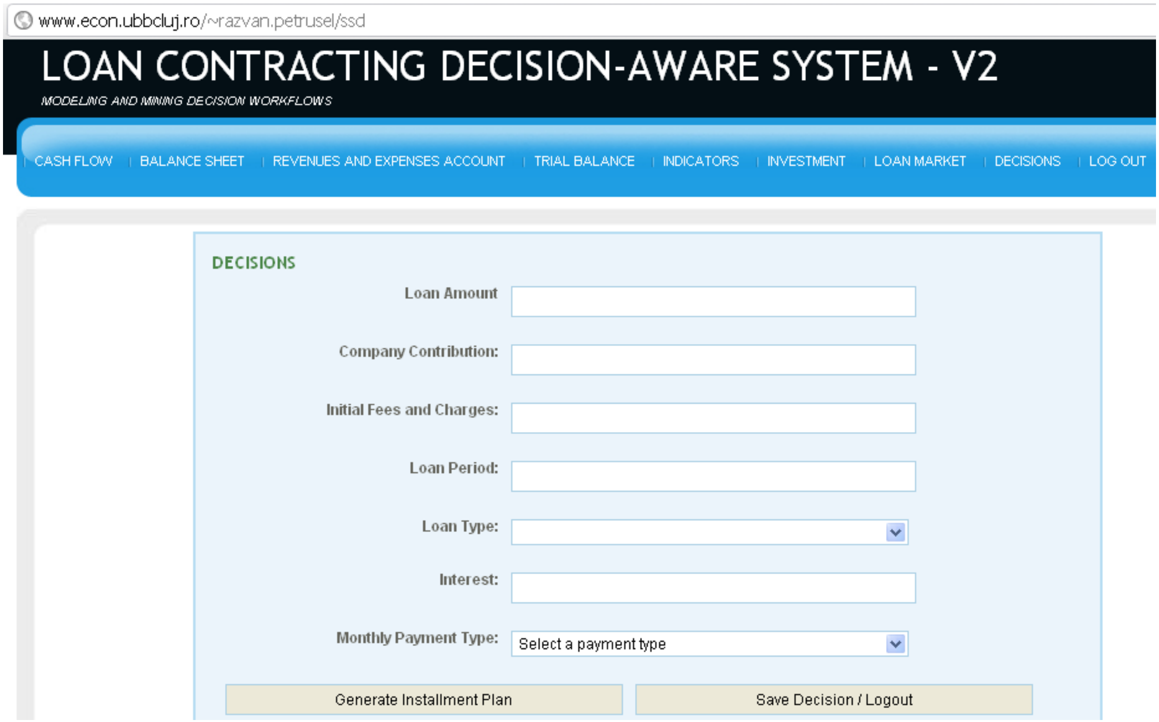


Fig. 10. Decision variables to be filled-in for the Loan Contracting Decision

When building and evaluating decision alternatives the user can review:

- all the financial data of the company (found in one of the forms: cash flow, balance sheet, income statement, trial balance and indicators) - see Figure 10;

- all information regarding the available loans on the market such as interest rate, commissions, minimum down payment, maximum loan value, required warranties, etc. (can be found in the menu Market Situation);
- the data about the investment to be made (provided in the Investment Evaluation menu).

One can notice that, with respect to the decision making process, the simulation model is based on four clusters of activities: knowledge accumulation on the situation of the company; about funding opportunities through loans, on the planned investment and on the generation / selection of decision alternatives. The expert's contribution towards the simulation model was, for this scenario, identifying data and information required and their sources. We also required the experts to provide some insights about data requiring aggregation. Basically, our research approach is to put the decision-maker in a well delimited decision making situation and provide him with all the necessary data items for reasoning during the decision-making process. Our goal is to capture, as accurate as possible, the undirected and unrestricted reasoning process.

The data shown in cash flow, balance sheet, income statement and trial balance forms are structured according to current laws and according to accounting and financial reporting standards. These forms have a similar structure as the one used in real financial statements. Scenario data is available for all the fields in each form. Of course, the data for the decision-making process ranges from not important to essential. This presentation has two advantages for this decision making situation:
- does not direct, in any way, the decision process of the user,
- the user needs to choose, from all the available data items, only the ones he finds relevant for this particular process.

For assessing the lending opportunities on the market, we use the same principle of providing a multitude of data items from which the user needs to select only the ones he finds relevant.

The logging of user's reasoning process is currently limited to a low-level approach. This means we can track, and record in the logs only the values that have been selected by the user for reviewing. The logging steps are:
- when a page is displayed all text boxes are empty (a) – Figure 11
- if the user wants to see a certain data item he needs to click on the appropriate text box (b) and the value is shown (c) – Figure 11
- the system logs in the tables of the database information such as text box's name, the value displayed in it, the timestamp, etc.. – Figure 12.

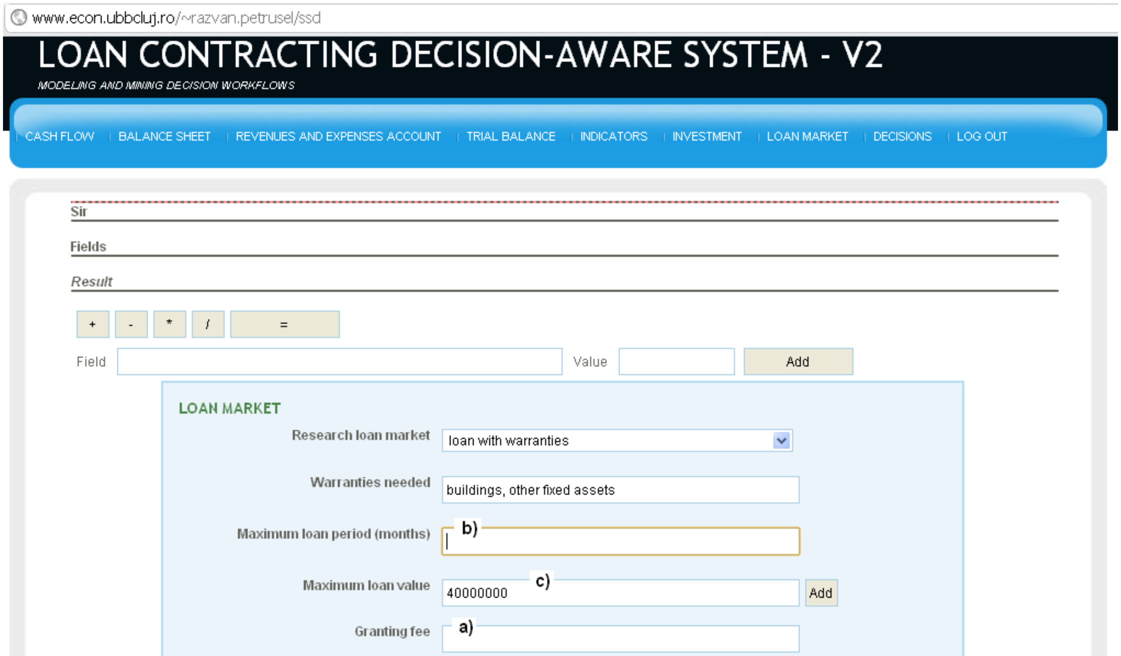
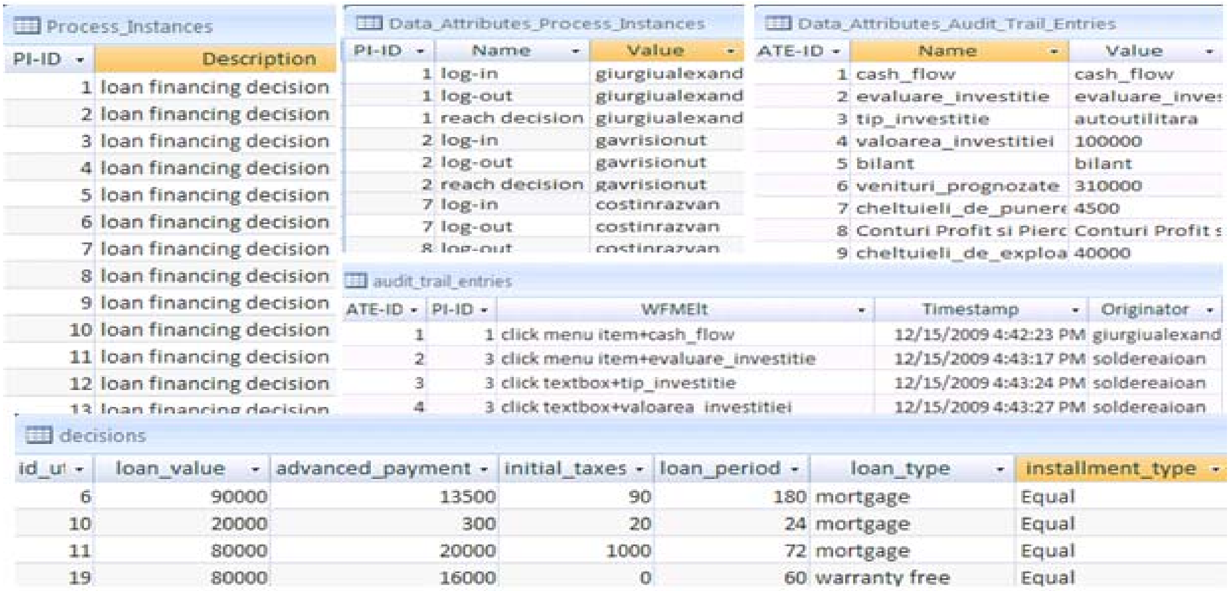


Fig. 11. User activity sequence example in the Loan Contracting Module



Process_Instances

PI-ID	Description
1	loan financing decision
2	loan financing decision
3	loan financing decision
4	loan financing decision
5	loan financing decision
6	loan financing decision
7	loan financing decision
8	loan financing decision
9	loan financing decision
10	loan financing decision
11	loan financing decision
12	loan financing decision
13	loan financing decision

Data_Attributes_Process_Instances

PI-ID	Name	Value
1	log-in	giurgiualexand
1	log-out	giurgiualexand
1	reach decision	giurgiualexand
2	log-in	gavrisionut
2	log-out	gavrisionut
2	reach decision	gavrisionut
7	log-in	costinrazvan
7	log-out	costinrazvan
8	log-out	costinrazvan

Data_Attributes_Audit_Trail_Entries

ATE-ID	Name	Value
1	cash_flow	cash_flow
2	evaluare_investitie	evaluare_inves
3	tip_investitie	autoutilitara
4	valoarea_investitiei	100000
5	bilant	bilant
6	venituri_prognozate	310000
7	cheltuieli_de_punere	4500
8	Conturi Profit si Pierc	Conturi Profit s
9	cheltuieli_de_exploa	40000

audit_trail_entries

ATE-ID	PI-ID	WFMElt	Timestamp	Originator
1	1	click menu item+cash_flow	12/15/2009 4:42:23 PM	giurgiualexand
2	3	click menu item+evaluare_investitie	12/15/2009 4:43:17 PM	soldereaioan
3	3	click textbox+tip_investitie	12/15/2009 4:43:24 PM	soldereaioan
4	3	click textbox+valoarea_investitiei	12/15/2009 4:43:27 PM	soldereaioan

decisions

id_ut	loan_value	advanced_payment	initial_taxes	loan_period	loan_type	installment_type
6	90000	13500	90	180	mortgage	Equal
10	20000	300	20	24	mortgage	Equal
11	80000	20000	1000	72	mortgage	Equal
19	80000	16000	0	60	warranty free	Equal

Fig. 12. User activity data stored in the log tables - example

The log data in Figure 12 can be interpreted as follows:

- in a usage session (eg. Process_Instances table for a new session, the value in the PI-ID field is 1), a decision process that was logged is bank loan financing. If several applications are logging different decision-making processes the Description field will allow us to clasify the actions to the right process;
- there may be complete processes (e.g. for PI-ID = 1, there are 3 entries in the table Data_Attributes_Process_Instances: Log-in, Log-out and Reach Decision). We also logged the user name. An example of an incomplete process is PI-ID = 7 session;
- in the Audit_Trail_Entries table the actual user activities are logged (e.g. for the session PI-ID = 3, the third activity is carried out by the user clicking on an a text box in case which shows that the user wanted to find the value related to the type of investment);
- in the Data_Attributes_Audit_Trail_Entries table we can see that for the activity 3 (ATE-ID = 3) the value actually shown to the user

in the text box (type invested) was "utility vehicle";

- in the Decisions table we can see the values inputted by the users for each decision variable for all the complete decision processes.

The result of decision mining is a decision workflow reference model based on the user behavior during the interaction with the decision-aware system. For this particular decision, loan contracting, the workflow model included in the figure below (Fig. 13) is extracted using the proposed approach from a log containing a total of 18 complete traces. The model-building method is semi-manual (after the completion of the testing algorithm an application that automatically extracts the model will be built).

One can notice that in the model shown in Figure 13a there are six clusters of activities. This grouping of user's activities is not surprising, considering the specific decision studied. Each cluster is composed of atomic activities (which is not mandatory for more complex decisions, when each cluster can be composed of multi-level sub-clusters). Basically, this model recommends steps to perform if a manager wants to make a decision on contracting a bank loan. As it can be seen, the focus is not the choice of a decision alternative but the actions that need to be performed by the decision maker during such a decision process. We argue that a clear an complete decision workflow is absolutely necessary in making the correct decision.

In Figure 13b we introduce the model produced by the algorithm Fuzzy Miner (included in the ProM Framework). In order to build this model we imported the logged decisional activities in ProM (using the ProM Import Tool) and ran a series of tests using the most important algorithms in the process mining field. The most conclusive results were obtained by the Fuzzy Miner. A short comparison reveals that a number of activities are identified both by DMA and Fuzzy Miner. However, DMA extracts a larger number of activities of those deemed essential by experts. However, DMA is not currently a general algorithm (therefore we expected it to produce better results on a customized log). In the next period we will create the necessary mathematical foundation of DMA and take the next steps towards generalization.

The model presented in Figure 13a is not a reference model since it was obtained based on the actions performed by students and master degree students from the Faculty of Economics and Business Administration at the University of Babes-Bolyai University in Cluj-Napoca. The sample is representative only to the students' knowledge and is not relevant to the real economy. The sample will be extended by future studies on groups of users from different industries and from different regions and countries (the application will be used by students from the Technical University of Eindhoven).

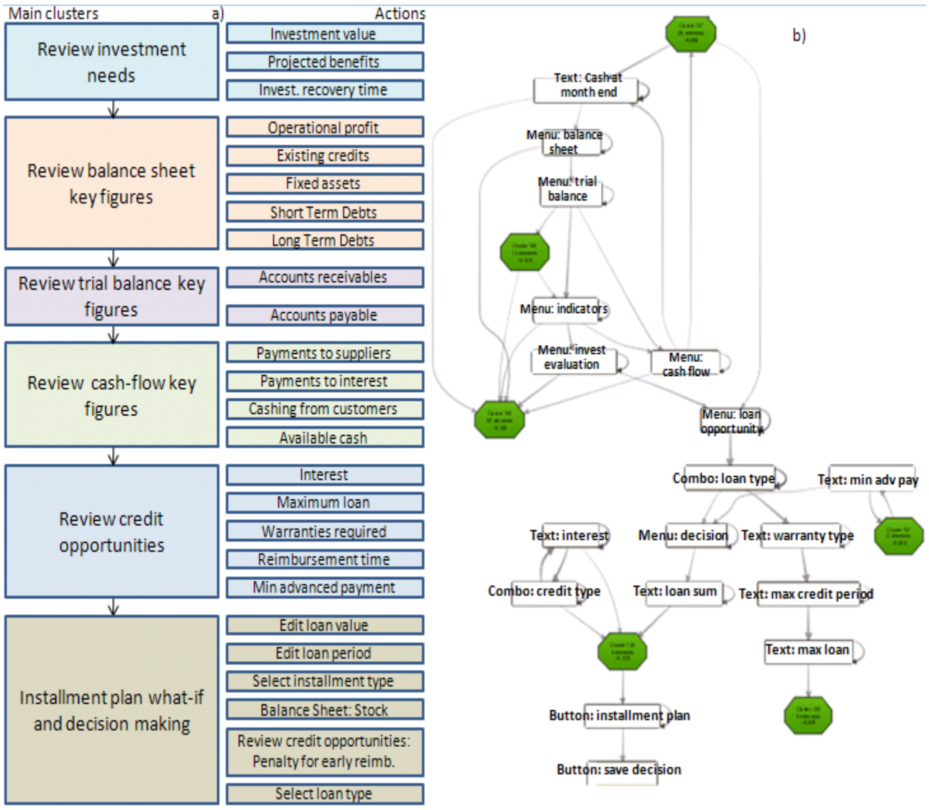


Fig. 13. Comparison of a) DMA and b) Fuzzy Miner models

The model is more relevant at the level of the local patterns of thinking. At this level an example is shown in Figure 14. The example details how the evaluation of alternatives was performed by four experts involved in our study. In Figure 14a we introduce a model of the interactions of the four experts. In Figure 14b we show the model extracted using DMA approach, highlighting the fact that verification of cash available is considered a key action, so it is "extracted" from the cluster.

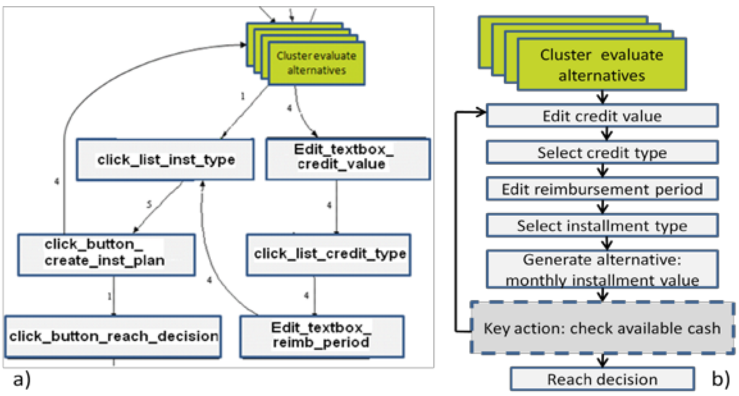


Fig. 14. Partial decision making pattern

4 Conclusions and Future Work

We introduced in this paper our approach towards implementing and using a decision-aware system. In the introduction we presented the general approach towards the researched area and our proposals regarding the architecture of an enterprise-wide decision aware system, the architecture of an individual module and the life cycle of a module.

There are several issues that need to be considered when designing such a system (the

decision that needs to be modelled, the activity logging techniques, the format of the logs). We presented in the paper the current implementation of a module of the decision-aware system that focuses on the loan contracting decision. The user is presented with all the data generated by the enterprise, with the available loans on the market and needs to decide on the amount to be loaned, on the loan period, on the type of the loan and on the type of instalments.
So far we only implemented the low-level logging techniques. This basically means that we log all the clicks performed by the user (the value of a data item is shown only when the text box is clicked). We also provide the user with the possibility to derive (calculate) new data based on the values in the simulation. We presented in more detail how logging is performed and how we actually log all the calculation steps performed by the user when deriving new data. All those actions are stored in five log tables that can be mined.
The mining approach presented in the paper is a framework that argues how we plan to extract new knowledge from the activity logs created before. There needs to be a balance between abstraction and completeness. In order to solve this problem we propose a multi-level model containing clusters and key actions.
In conclusion, we believe that this approach can be an alternative to traditional methods of extracting the knowledge regarding the decision-making processes, without trying to replace them. So far, this approach is an ongoing research. But prospects are promising, considering the results outputted by the proof-of-concept implementation.

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

References

[1] M.T. Wynn, M. Dumas, C.J. Fidge, A.H.M. ter Hofstede, and W.M.P. van der Aalst "Business Process Simulation for Operational Decision Support" *Lecture Notes in Computer Science BPM 2007*, vol. 4928, pp. 66-77, 2008.
[2] W.M.P. van der Aalst, J. Nakatumba, A. Rozinat, and N. Russell „Business Process Simulation: How to get it right?" in *International Handbook on Business Process Management*, Vol. 1, pp. 317-342, 2009.
[3] A. Rozinat, R.S. Mans, M. Song, and W.M.P. van der Aalst „Discovering Simulation Models", *Information Systems*, Vol. 34, No. 3, pp. 305-327, 2009.
[4] A. Rozinat, M. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and C. Fidge. „Workflow Simulation for Operational Decision Support" *Data and Knowledge Engineering*, Vol. 68, No. 9, pp. 834-850, 2009.
[5] B.F. van Dongen, A.K. Alves De Medeiros, and L. Wen, „Process mining : overview and outlook of Petri net discovery algorithms" *Transactions on Petri Nets and Other Models of Concurrency*, pp. 225-242, 2009.
[6] T. Weijters and W.M.P. van der Aalst "Process Mining: Discovering Workflow Models from Event-Based Data" in Proc. *The 13th Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2001)*, 2001, pp. 283-290.
[7] J.E. Cook and A.L. Wolf "Discovering Models of Software Processes from Event-Based Data", *ACM Transactions on Software Engineering and Methodology*, Vol. 7, No. 3, pp. 215-249, 1998.
[8] J. Herbst and D. Karagiannis, "Integrating Machine Learning and Workflow Management to Support Acquisition and Adaptation of Workflow Models,", In Proc. *9th International Workshop on Database and Expert Systems Applications (DEXA'98)*, 1998, pp.745-757.
[9] W.M.P. van der Aalst and K. M. van Hee, *Workflow management: models, methods, and systems*, Cambridge, MA: MIT Press, 2002.
[10] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros,

"Workflow Patterns", *Distributed and Parallel Databases*, vol. 14, no. 1, pp. 5-51, 2003.

[11] R. Petruşel, "Collaborative Virtual Enterprise Environment and Decision Mining", *Informatica Economică Journal*, vol. 13. no. 2, pp. 59-68, June 2009.

[12] T. Reenskaug, *The original MVC reports*, Xerox PARC technical note, May 1979.

Răzvan PETRUŞEL holds a Ph.D. in Cybernetics, Statistics and Business Informatics starting 2008. He started in 2003 as a full-time Ph.D. student at the Business Information Systems Department, Economical Sciences and Business Administration Faculty, in Babeş-Bolyai University of Cluj-Napoca. In 2007 he became an assistant professor and since 2009 he holds the current position as lecturer. His research is focused on DSS Specification, Modeling and Analysis; Process Mining; Workflow Management; and Decision Mining and Analysis.

Daniel MICAN is a PhD student in Business Information Systems at Babeş-Bolyai University of Cluj-Napoca. His main research areas are: Web Portals, Web-based Applications, Content Management Systems, Personalisation and Recommender Systems, Search Engine Optimization and Internet Marketing Services. He is Teaching Assistant at Faculty of Economics and Business Administration, Babeş-Bolyai University of Cluj-Napoca.

Cristina-Claudia DOLEAN has graduated the Faculty of Economics and Business Administration, Babeş Bolyai University, Cluj-Napoca in 2008. She holds a bachelor degree in Business Informatics and a master degree in E-Business. She is currently a PhD student in the field of Business Informatics. Her current research interest include Process mining and Workflow Management.