Towards Business Process Model Extension with Cost Perspective Based on Process Mining
Petri Net Model Case

Dhafer Thabet, Sonia Ayachi Ghannouchi and Henda Hajjami Ben Ghézala
RIADI Laboratory, National School of Computer Sciences, Mannouba University, Mannouba, Tunisia

Keywords: Business Process Management, Business Process Improvement, Process Mining, Petri Net Model Extension, Cost Information.

Abstract: Business process improvement has ever been the major organizations concern to enhance efficiency, flexibility and competitiveness. Business process management is a contemporary approach which includes different techniques to support business process improvement along its lifecycle phases. Process mining is a maturing technology based on event logs analysis giving insight on what is really happening at the operational level. Process model extension is one of the three process mining types, which provides different perspectives of the business process. Otherwise, cost perspective is one of the relevant information needed by organization managers to take the suited process improvement decisions. In this paper, we propose an approach allowing Petri Net model extension with cost information using process mining extension technique. Besides, we present the main details about its implementation and the test case results.

1 INTRODUCTION

Efficiency, flexibility and competitiveness have ever been the main concerns of all organizations (Briol, 2008). In this context, Business Process Management (BPM) is a contemporary approach which has been highly considered for its potential of enhancing organizations business processes continuously (Adams, et al., 2010; Rosemann and vom Brocke, 2010). In our work, we consider the overall goal of supporting organizations decision makers to enhance their business processes.

Among the different BPM techniques, process mining is used to analyze business processes based on event logs recorded by the systems executing these processes (Adams, et al., 2010; Rosemann and vom Brocke, 2010) which makes its results relevant and accurate for managers and decision makers. Event logs can be used to conduct three types of process mining (van der Aalst, 2011; IEEE Task Force on Process Mining, 2012; Rozinat, 2010): (1) discovery: produces a business process model using event logs; (2) conformance: an existing process model is compared with the corresponding event logs to identify the eventual deviations; (3) enhancement: includes two sub-types: repair (improving the model to better reflect reality) and extension (using information extracted from event logs to enrich the model). The extension technique enables process model enhancement with different perspectives such as the organizational, the case and the time perspectives.

Otherwise, organizations are always concerned with reduction of costs incurred during the execution of their business processes. Management accounting is the field dealing with how cost and other information should be used for planning, controlling, continuous improvement and decision making (Weygandt, Kimmel and Kieso, 2010; Hansen and Mowen, 2006). It includes several techniques such as: Activity-Based Costing/Management (ABC/M) (Weygandt, Kimmel and Kieso, 2010; Hansen and Mowen, 2006); Time-Driven ABC (TDABC) (Kaplan and Anderson, 2003); and Resource Consumption Accounting (RCA) (RCA Institute, 2008; Clinton and Webber, 2004). The goal of these techniques is to measure costs incurred during process execution and to allocate them to the business process operations.

These cost information would be easier to be interpreted and more significant and accurate for decision makers if they are associated to the

Thabet D., Ayachi Ghannouchi S. and Hajjami Ben Ghézala H..
Towards Business Process Model Extension with Cost Perspective Based on Process Mining - Petri Net Model Case.
DOI: 10.5220/0004883003350342
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 335-342
ISBN: 978-989-758-029-1

corresponding elements of the business process model. This leaded us to consider the issue of business process model extension with cost information based on process mining extension technique. As a first step of our work, we considered to start by dealing with the case of Petri Net models. In this paper, we propose an approach as well as its implementation for cost extension of Petri Net models using the process mining extension technique.

In the remainder of this paper, we present the research questions of our work in Section 2. In Section 3, we give an overview about related works. Section 4 presents the basic concepts related to Petri Nets. Section 5 deals with the description of our proposed approach. Section 6 presents the implementation of the proposed approach. Finally, in Section 7, we summarize our contribution and present our future works.

2 RESEARCH QUESTIONS

On one hand, the process mining extension technique enriches the business process model with a wealth of information about the operational level based on event logs. On the other hand, cost information related to the business process is crucial and interesting for decision making support. Therefore, the main research question of our work is how to extend the business process model with cost information using the process mining extension technique. This leads us to deal with the two following questions:

- (1) Is it possible to provide cost information from event logs? If so, how?
- (2) How to extend the business process model with the obtained cost information?

The question (1) is about the possibility and then the way for collecting cost information from event logs. If we assume that it is possible to do so, the question (2) deals with the manner by which the business process model could be extended with the collected cost information. In this paper, as a starting point in our work, we propose a solution for the first research question (1) while we start by dealing with the second research question (2) for business processes modelled with Petri Nets. Therefore, the main contribution of this paper is the proposal of an approach as well as its implementation providing accurate cost information as a perspective of Petri Net models using the process mining extension technique.

3 RELATED WORKS

Nauta (2011) proposed an architecture to support cost-awareness in process mining and its first implementation. The architecture contains two parts: (1) in the management accounting part, the accountant provides cost drivers and cost functions. A cost driver defines how cost is associated with a process and the cost amount incurred per unit of activity or time. A cost function defines the content of a cost report. (2) The Business Process Management System (BPMS) part consists of 3 main steps. (a) In the first step, cost drivers and functions, the business process model and the organizational model are taken as inputs to create a cost model (Nauta, 2011). (b) During the second step, event logs together with the cost model are used to calculate cost information that is used to annotate the initial event log which is in the XES (eXtensible Event Stream) format. XES is an XML-based generic format for event logs (Nauta, 2011; Wynn, Low and Nauta, 2013). A XES event log consists of an arbitrary number of traces. Each trace contains an arbitrary number of events (Hverbeek, 2012). The cost annotation is performed, per cost type, in the final event of each task instance. The obtained cost annotated event log is also in the XES format. (c) In the third step, the cost annotated event log is used to create cost reports (Nauta, 2011).

The context of this work is a current PhD project named "Cost-aware BPM" started in 2012 (Wynn, 2012). The author proposes a framework to support management accounting decisions on cost control for monitoring, predicting and reporting based on the work of Nauta (2011). The implemented plug-ins of this framework allow the generation of different cost reports (Wynn, Low and Nauta, 2013).

4 PETRI NETS

4.1 Definitions

A Petri Net is a directed bipartite graph populated by three types of objects: places, transitions, and directed arcs. Directed arcs connect places to transitions or transitions to places (Li and Zhou, 2009; Wang, 2007). Places are graphically represented by circles and transitions by boxes or bars. A place can hold tokens denoted by black dots, or a positive integer representing their number. The distribution of tokens over the places is called a marking which corresponds to a state of the modelled system.

The initial token distribution is called the initial marking. A Petri Net is formally defined as a 5-tuple $N = (P, T, I, O, M_0)$, where:

- $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places;
- $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions, $P \cup T \neq \emptyset$, and $P \cap T = \emptyset$;
- $I: P \times T \rightarrow N$ is an input function that defines directed arcs from places to transitions, where N is a set of non-negative integers;
- $O: T \times P \rightarrow N$ is an output function that defines directed arcs from transitions to places;
- $M_0: P \rightarrow N$ is the initial marking.

4.2 Properties

Petri Nets have a number of properties which allow the system designer to identify the presence or absence of the application domain specific functional properties of the considered system. Two types of properties can be distinguished: behavioural and structural properties. The behavioural properties depend on the initial marking of a Petri Net while the structural ones don't depend on it. The most important behavioural properties are : reachability, boundedness, safeness, liveness, reversibility (Li and Zhou, 2009; Wang, 2007).

4.3 Types

There are a number of types of Petri Nets which may be classified in three levels (Trompedeller, 1995): (1) Petri Nets characterised by places which can represent boolean values (examples: Elementary Net Systems, State Machines); (2) Petri Nets characterised by places which can represent integer values (examples: Marked Graphs, Free Choice Nets); (3) Petri Nets characterised by places which can represent high-level values (Examples: High-Level Petri Nets, Coloured Petri Nets).

5 PROPOSED APPROACH

The proposed solution is an approach allowing the extension of Petri Net models with cost perspective using process mining extension technique. In the following, the approach overview and the Petri Net meta-model cost extension are presented.

5.1 Approach Overview

As shown in Figure 1, the inputs of our proposed approach are: the Petri Net, that models the business process, and the event log which was generated by the BPMS executing this process and then annotated with cost data by the application of Nauta's approach (2011). The main step of our proposed approach is the extension of the Petri Net with cost information. This step consists of three main sub-steps: (1) In the first sub-step, the cost annotated event log is used to extract cost information. (2) During the second sub-step, the Petri Net model is loaded into memory. (3) In the third sub-step, cost information is computed with respect to the user's requirements and is associated to the corresponding elements of the Petri Net. The output of this step is a Petri Net extended with cost information. Finally, the Petri Net is displayed along with the associated cost perspective.

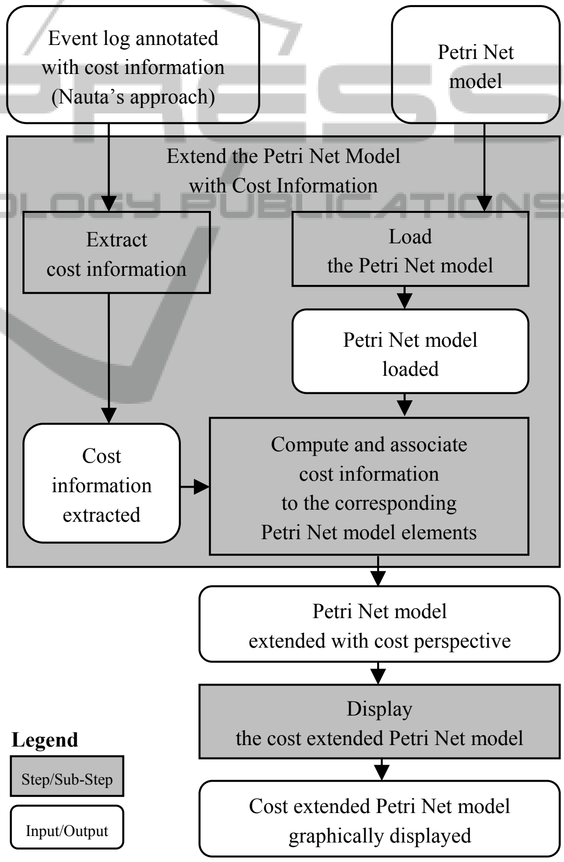


Figure 1: Overview of our proposed approach.

5.2 Petri Net Meta-Model Extension with Cost Data Structure

Because of the different types of Petri Nets, several works focused on the proposal of a standard specification for Petri Net models to facilitate their transfer between different tools.

The Petri Net Markup Language (PNML) is a

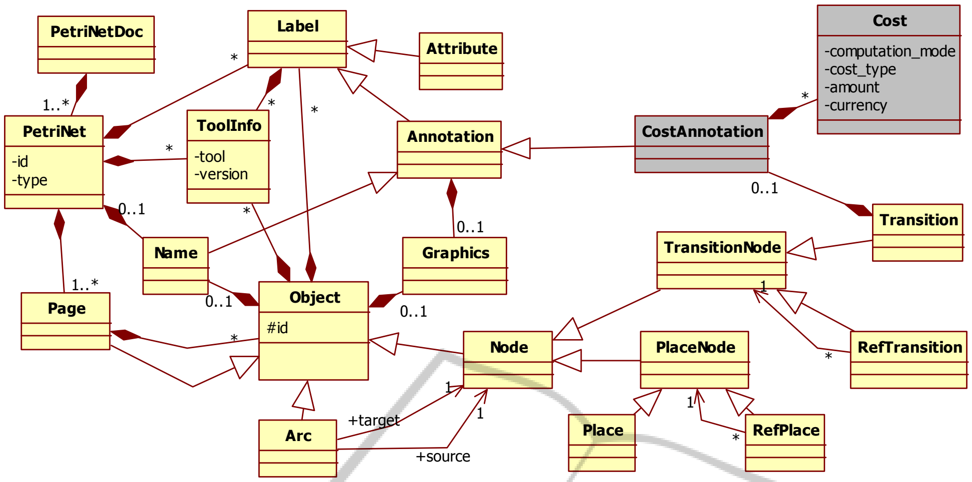


Figure 2: PNML core model extended with Cost data structure.

proposal of an XML-based interchange format for Petri Nets and has been defined as an international standard (ISO/IEC 15909 series) which defines a meta-model of four packages: PNML core model, Place/Transition Nets, Symmetric Nets and High-level Petri Net Graphs. The last three packages are related to specific versions of Petri Nets while the first one can represent any kind of Petri Net (Billington, et al., 2003; LIP6, 2012; 2013; Hillah, et al., 2009). Therefore, in our work, we consider the PNML core model package as the Petri Net meta-model. Figure 2 shows the UML (Unified Modeling Language) class diagram representing the PNML core model extended with the cost data structure. A document that meets the PNML core model is a Petri Net document which contains one or more Petri Nets. Each Petri Net has a name and contains one or more pages. Each page is an object that consists of objects. Each object has a unique identifier, a name and may have graphical information. Besides, an object may be a node or an arc. An arc must connect nodes of the same page. A node can be whether a transition node or a place node. A transition node is whether a transition or a reference transition. Similarly, a place node is whether a place or a reference place. Reference transitions and reference places are used to allow the connection of nodes that belong to different pages, by an arc. A Petri Net, its pages and its objects can have labels which may be annotations or attributes. For some tools, specific information might be stored for each Petri Net, object or label (Hillah, et al., 2009).

The cost data structure consists of the two grey colored classes (CostAnnotation and Cost classes) and their corresponding relationships as shown in Figure 2. The Cost class represents the cost concept which is described by: computation mode, type, amount (value) and currency. The computation mode defines how the cost amount is calculated (average, maximum or minimum). The cost type is one of the types defined in the cost annotated event log: fixed cost, labor cost and overhead cost. Besides, we added a fourth cost type: the total cost (sum of all the other cost types values). The cost annotation is a kind of annotation as it represents information textually displayed. Furthermore, a cost annotation consists of a set of costs and is associated to a transition as costs are attributed to tasks in the cost annotated event log. Moreover, the cost information of a reference transition is that associated to its corresponding transition.

6 APPROACH IMPLEMENTATION

In order to test our approach, we implemented a prototype allowing the extension of a Petri Net model with cost information extracted from the corresponding cost annotated event log. Implementation and test details are presented in the next sub-sections.

6.1 Tool Architecture Overview

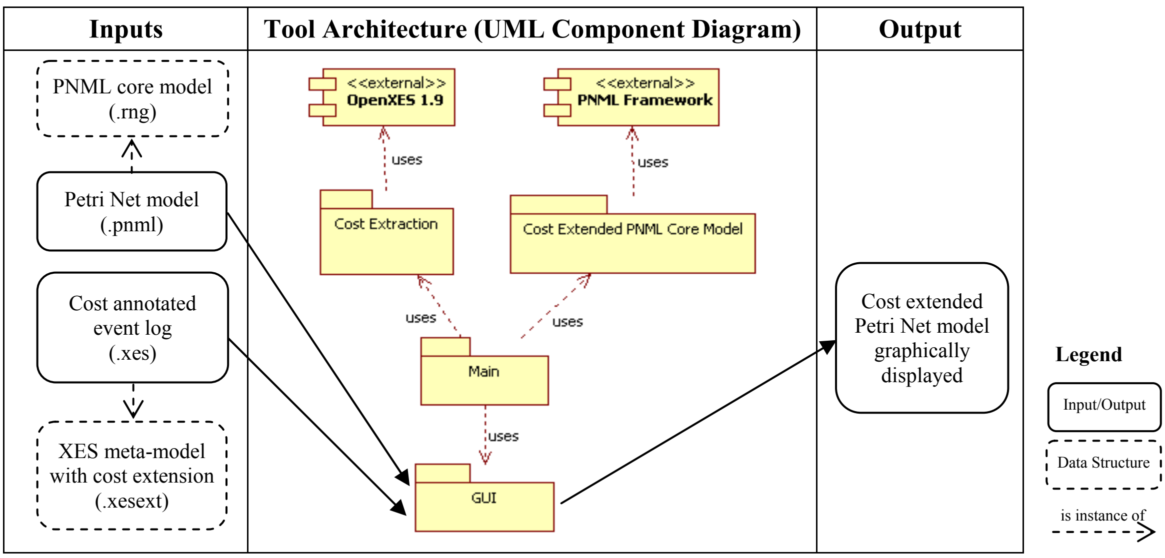
As shown in the left side column of Table 1, the required inputs for our tool are the following: (1) the Petri Net model file (PNML format) which meets the PNML core model structure; and (2) the corresponding cost annotated event log (XES format) which is produced with respect to the cost extended XES meta-model (Nauta, 2011; Wynn, Low and Nauta, 2013). The central part of Table 1 illustrates the internal structure of the implemented tool which is modelled by an UML component diagram. The inputs are imported using graphical user interfaces (GUI package). The extraction of cost information from the cost annotated event log is performed using the cost extraction package that imports the OpenXES (1.9 version) library which is a reference implementation of the XES standard for storing and managing event log data (Hverbeek, 2012). Besides, the Petri Net model is loaded using the cost extended PNML core model package which uses the PNML framework that is a prototype implementation of the international standard on Petri Nets (LIP6, 2012; 2013). In addition to the orchestration of these functionalities, the other roles of the main package are the handling of the extracted cost data, their association to the corresponding loaded Petri Net elements and the display of the cost extended Petri Net model using the GUI package. Then, the output produced by the tool is a Petri Net model extended with a cost perspective graphically displayed.

6.2 Cost Extraction and Extension Algorithms

Figure 3 presents the algorithm of cost information extraction from cost annotated event log. When starting the parse of the cost annotated event log, a log cost object (lco) is initialized. For each trace in the event log, a trace cost object (trco) is defined. Each time an end event is found in the current trace, a task instance cost object (tico) is initialized. For each cost type element found in the current end event, the related cost data is extracted and used to create an elementary cost object (eco) which will be added to the current tico. Cost objects related to task instances of one same task in one same trace are grouped within a task cost object (taco). Once the current trace is fully parsed, the current trco is added to the current lco.

Once the cost data is extracted from the cost annotated event log and the Petri Net model is loaded with respect to the PNML core model, the model extension sub-step is performed. Figure 4 shows the Petri Net model cost extension algorithm. As the cost extracted data is related to tasks in the cost annotated event log, we consider simply the extension of transitions contained in each page of the Petri Net. A cost annotation object (cao) is initialized each time a transition is found in the current page. The cao will contain a list of cost objects. For each combination of the computation modes and the cost types, selected by the user, a cost object is created with the corresponding data (computation mode, type, amount and currency) and then it is added to the current cao. After that, the cao is added in turn to the current transition object. Thus, these instructions are executed for each transition of each page of the Petri Net.

Table 1: General architecture of the Petri Net model cost extension tool.



```
Initialize a log cost object
For each trace in the event log
  Initialize a trace cost object
  For each end event in the trace
    Initialize a task instance cost object
    For each cost type in the end event
      Create an elemental cost object using the extracted cost information
      Add the elemental cost object to the task instance cost object
    End For
    If the corresponding task cost object exists in the trace cost object Then
      Add the task instance cost object to the corresponding task cost object
     Else
      Create the corresponding task cost object
      Add the task instance cost object to the created task cost object
      Add the created task cost object to the trace cost object
    End If
  End For
  Add the trace cost object to the log cost object
End For
```

Figure 3: Algorithm of the cost information extraction.

```
For each page in the Petri Net
  For each transition in the page
    Initialize a cost annotation object
    For each computation mode selected by the user
      For each cost type selected by the user
        Create and Add corresponding cost object to the cost annotation object
      End For
    End For
    Add the cost annotation object to the current transition object
  End For
End For
```

Figure 4: Algorithm of the Petri Net model cost extension.

6.3 Test Example

In order to test our approach, we considered the case of a simple phone repair process. Figure 5 shows the Petri Net model representing the considered business process. The process begins by the registration of the broken phone and then it is analyzed to determine the defect type. Depending on the severity of the defect, a simple repair or a complex repair is carried out. Then, the phone is tested to check whether it is fixed. If so, the repair details are archived and it is returned to the customer. If it is not fixed, the repair and then the test are restarted again. If the phone is still broken after the fifth repair test, the repair details are archived and the phone is returned to the customer. Otherwise, the customer is informed about the defect type after the defect analysis and before archiving the repair details (Nauta, 2011; Process Mining Group, 2013). This process example was already used as a test case example in Nauta's approach. We used the produced cost annotated event log together with its corresponding Petri Net model as inputs in our tool test phase.

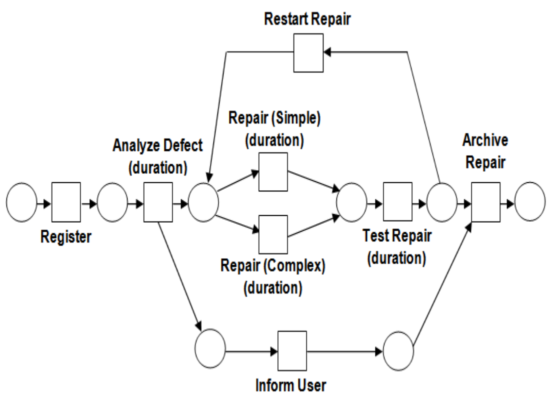


Figure 5: Petri Net model of the simple telephone repair process (Nauta, 2011, p.14).

6.4 Test Results

After launching the Petri Net model cost extension tool, the user selects the cost extension parameters (Figure 6) including the cost computation mode(s), the cost type(s), the PNML file (Petri Net model) and the XES file (cost annotated event log).

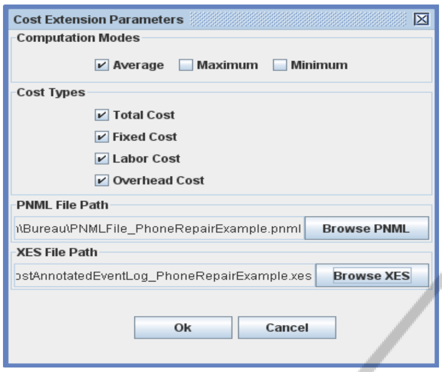


Figure 6: Selection of the cost extension parameters.

The user selects one or more of the provided computation modes: average, maximum and minimum. Figure 6 shows that the user selected to compute the average cost. The user also selects one or more of the provided cost types: fixed, labor, overhead and total. As shown in Figure 6, the user selected to get all the provided types of costs. The Petri Net model and the corresponding cost annotated event log files are selected by providing their absolute path. In Figure 6, the selected PNML and XES files correspond to the simple phone repair process example. Once the cost extension parameters are validated by the user, the cost extended Petri Net is displayed on the main tool frame (Figure 7). The cost information is displayed for each transition using tooltips appearing when the user hovers the mouse over one transition which will be automatically highlighted. The cost information related to a transition is described using a table which entries represent the different combinations between the computation mode(s) and the cost type(s) selected by the user. Each entry presents the corresponding cost value and currency. For instance, as the user selected to get an average value for each cost type (Figure 6), each transition of the selected Petri Net model will be extended with a table containing an average value of each selected cost type. Figure 7 shows the cost information related to the "Analyze Defect" transition which includes an average total cost of 17.38 AUD (Australian Dollar), an average fixed cost of 8.20 AUD, an average labor cost of 3.10 AUD and an average overhead cost of 6.07 AUD.

7 CONCLUSION AND FUTURE WORKS

In this paper, we proposed an approach for Petri Nets extension with cost information using process mining extension technique. Besides, we proposed a cost extension of the general Petri Net meta-model (PNML core model) to store the cost information extracted from the cost annotated event log produced by Nauta's approach (2011). Furthermore, we described the implementation details of our Petri Net cost extension tool which was tested for the case of a simple phone repair process.

For this moment, the implemented prototype allows decision makers to have detailed cost information directly related to the Petri Net which helps them better understand their business process

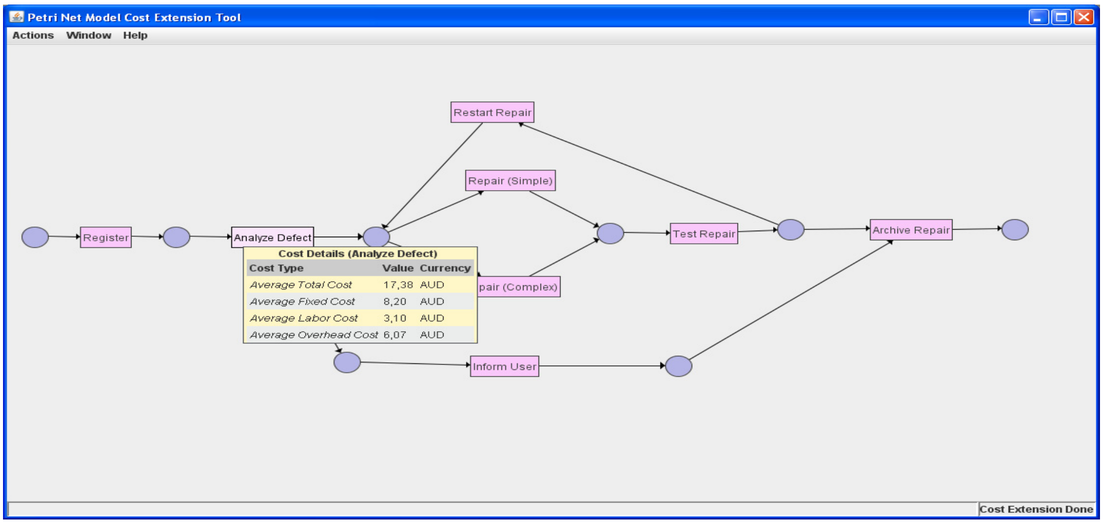


Figure 7: Displaying the cost information related to the "Analyze Defect" transition.

from an accurate cost point of view which in turn, may lead them to detect the eventual cost problems in their business process. Besides, detection of cost problems in the business process is a mandatory phase to be able to take the suited actions for business process enhancement in terms of cost. Therefore, the main current contribution of our work is the assistance of decision makers in the first phase of business process enhancement in terms of cost.

Currently, we are planning to conduct surveys with management accounting and business process management experts to improve and deepen the cost information handling (structuring, analysis and presentation) in order to make it more suitable and useful for decision making. In future works, we consider the generalization of the proposed approach to support cost extension of any business process model notation (not only Petri Nets). Finally, real world case studies will be carried out in order to evaluate our approach and tool performances.

REFERENCES

Adams, M., Russell, N., ter Hofstede, A. H. M. and van der Aalst, W. M. P., 2010. *Modern Business Process Automation*, Springer. Heidelberg.

Billington, J., Christensen, S., van Hee, K., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C. and Weber, M., 2003. The Petri Net Markup Language: Concepts, Technology, and Tools. In *ICATPN'03, 24th International Conference on Applications and Theory of Petri Nets*, Springer.

Briol, P., 2008. *Ingénierie des Processus Métiers: de l'élaboration à l'exploitation*, Briol Patrice.

Clinton, D. and Webber, S. A., 2004. RCA at Clopay: Here's innovation in management accounting with Resource Consumption Accounting. In *Strategic Finance*.

Hansen, D. R. and Mowen, M. M., 2006. *Cost Management: Accounting and Control*, Thomson South-Western. United States of America.

Hillah, L. M., Kindler, E., Kordon, F. and Petrucci, L., 2009. A primer on the Petri Net Markup Language and ISO/IEC 15909-2*. In *CPN'09 10th International workshop on Practical Use of Colored Petri Nets and the CPN Tools*. Denmark.

Hverbeek, 2012. *Extensible Event Stream*. [online] Available at: <http://www.xes-standard.org> [Accessed June 2013].

IEEE Task Force on Process Mining, 2012. Process Mining Manifesto. In *BPM 2011 Workshops*. Springer.

Kaplan, R.S. and Anderson, S.R., 2003. Time-Driven Activity-Based Costing. In *HBS Working Paper*.

Li, Z. and Zhou M., 2009. Petri Nets. In: *Deadlock Resolution in Automated Manufacturing Systems*. Ch.2. Springer. (online) Available at: Springer website <http://www.springer.com/computer/swe/book/978-1-84882-243-6> (Accessed July 2013).

LIP6 (Laboratoire d'Informatique de Paris 6), 2012. *PNML Framework download site*. (online) Available at: <http://pnml.lip6.fr> (Accessed July 2013).

LIP6 (Laboratoire d'Informatique de Paris 6), 2013. *PNML reference site*. (online) Available at : <http://www.pnml.org> (Accessed July 2013).

Nauta, W.E., 2011. *Towards Cost-Awareness in Process Mining*. Master's thesis. Eindhoven. (online) Available at: <http://alexandria.tue.nl/extra1/afstversl/wsk-i/nauta2011.pdf> (Accessed April 2012).

Process Mining Group, 2013. *Process Mining: Research, Tools and Application*. (online) Available at: <http://www.processmining.org> (Accessed June 2013).

RCA Institute, 2008. *Resource Consumption Accounting Institute*. (online) Available at: <http://www.rcainstitute.org> (Accessed April 2012).

Rosemann, M. and vom Brocke, J., 2010. *Handbook on Business Process Management 1*, Springer. Heidelberg.

Rozinat, A., 2010. *Process mining: Conformance and Extension*, PhD Thesis, University Press Facilities. Eindhoven.

Trompedeller, M., 1995. *Petri Nets Classification.* (online) Available at: <http://www.informatik.uni-hamburg.de/TGI/PetriNets/classification/> (Accessed July 2013).

van der Aalst, W. M. P., 2011. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*, Springer-Verlag. Berlin.

Wang, J., 2007. Petri Nets for Dynamic Event-Driven System Modeling. In: P. Fishwick, ed. 2007. *Handbook of Dynamic System Modeling*. CRC Press. [online] Available at: Monmouth University website < http://bluehawk.monmouth.edu/~jwang/Petri%20Nets%20--%20Introduction.pdf> (Accessed July 2013).

Weygandt, J. J., Kimmel, P. D. and Kieso, D. E., 2010. *Managerial Accounting Tools for Business Decision Making*, John Wiley & Sons Inc. United States of America.

Wynn, M., 2012, *Dr. Moe Thandar Kyaw Wynn's homepage*. (online) Available at: <http://www.yawlfoundation.org/moe/index.html> (Accessed May 2012).

Wynn, M. T., Low, W. Z. and Nauta, W., 2013. A Framework for Cost-Aware Process Management: Generation of Accurate and Timely Management Accounting Cost Reports. In *CRPIT'13, Conferences in Research and Practice in Information Technology*. Australian Computer Society Inc.