Workflow Simulation for Operational Decision Support Using Design, Historic and State Information

A. Rozinat[1], M. T. Wynn[2], W. M. P. van der Aalst[1,2], A. H. M. ter Hofstede[2], and C. J. Fidge[2]

[1] Information Systems Group, Eindhoven University of Technology,
P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
{a.rozinat,w.m.p.v.d.aalst}@tue.nl

[2] Business Process Management Group, Queensland University of Technology,
GPO Box 2434, Brisbane QLD 4001, Australia.
{m.wynn,a.terhofstede,c.fidge}@qut.edu.au

Abstract. Simulation is widely used as a tool for analyzing business processes but is mostly focused on examining rather abstract steady-state situations. Such analyses are helpful for the initial design of a business process but are less suitable for operational decision making and continuous improvement. Here we describe a *simulation system for operational decision support* in the context of workflow management. To do this we exploit not only the workflow's *design*, but also logged data describing the system's observed *historic* behavior, and information extracted about the current *state* of the workflow. Making use of actual data capturing the current state and historic information allows our simulations to accurately predict potential near-future behaviors for different scenarios. The approach is supported by a practical toolset which combines and extends the workflow management system YAWL and the process mining framework ProM.

Keywords: Workflow Management, Process Mining, Short-term Simulation.

1 Introduction

Business process simulation is a powerful tool for process analysis and improvement. One of the main challenges is to create simulation models that *accurately* reflect the real-world process of interest. Moreover, we do not want to use simulation just for answering strategic questions but also for tactical and even operational decision making. To achieve this, different sources of simulation-relevant information need to be leveraged. In this paper, we present a new way of creating a simulation model for a business process supported by a workflow management system, in which we integrate design, historic, and state information.

Figure 1 illustrates our approach. We consider the setting of a *workflow system* that supports some *real-world process* based on a *workflow and organizational model*. Note that the workflow and organizational models have been

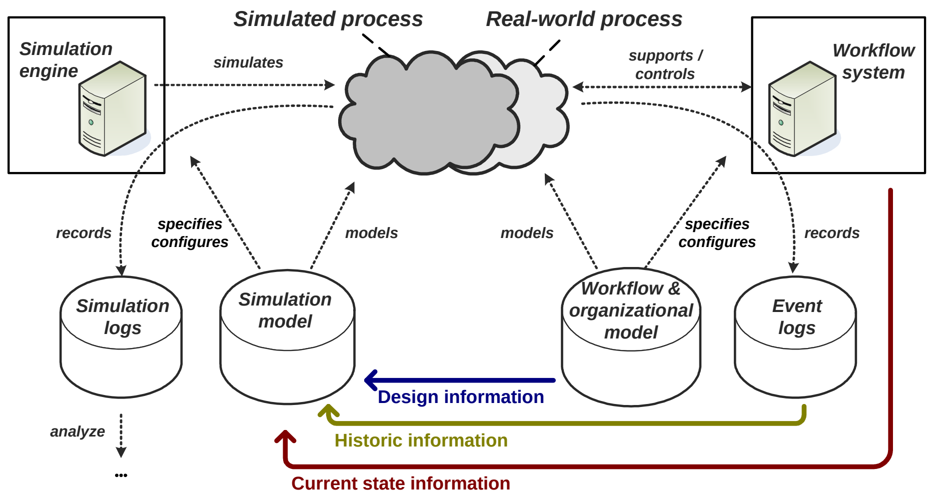


Fig. 1. Overview of our integrated workflow management (right) and simulation (left) system

designed before enactment and are used for the configuration of the workflow system. During the enactment of the process, the performed activities are recorded in *event logs*. An event log records events related to the offering, start, and completion of work items, e.g., an event may be 'Mary completes the approval activity for insurance claim XY160598 at 16.05 on Monday 21-1-2008'.

The right-hand side of Figure 1 is concerned with enactment using a workflow system while the left-hand side focuses on analysis using simulation. In order to link enactment and simulation we propose to use three types of information readily available in workflow systems to create and initialize the simulation model.

– *Design information.* The workflow system has been configured based on an explicit process model describing control and data flows. Moreover, the workflow system uses organizational data, e.g., information about users, roles, groups, etc.
– *Historic information.* The workflow system records all events that take place in 'event logs' from which the complete history of the process can be reconstructed. By analyzing historic data, probability distributions for workflow events and their timing can be extracted.
– *State information.* At any point in time, the workflow process is in a particular state. The current state of each process instance is known and can be used to initialize the simulation model. Note that this current state information includes the control-flow state (i.e., 'tokens' in the process model), case data, and resource data (e.g., resource availability).

By merging the above information into a simulation model, it is possible to construct an *accurate model based on observed behavior* rather than a manually-constructed model which approximates the workflow's anticipated behavior. Moreover, the state information supports a 'fast forward' capability, in which simulation can be used to explore different scenarios with respect to their *effect in the near future*. In this way, simulation can be used for *operational decision making*.

Based on this approach, the system design in Figure 1 allows different simulation experiments to be conducted. For the 'as-is' situation, the simulated and real-world processes should overlap as much as possible, i.e., the two process 'clouds' in Figure 1 coincide. For the 'to-be' situation, the observed differences between the simulated and real-world processes can be explored and quantified. In our implementation we ensure that the simulation logs have the same format as the event logs recorded by the workflow system. In this way we can use the *same tools* to analyze both simulated and real-world processes.

To do this, we need state-of-the art *process mining* techniques to analyze the simulation and event logs and to generate the simulation model. To demonstrate the applicability of our approach, we have implemented the system shown in Figure 1 using ProM [1] and YAWL [2]. YAWL is used as the workflow management system and has been extended to provide high-quality design, historic, and state information. The process mining framework ProM has been extended to merge the three types of information into a single simulation model. Moreover, ProM is also used to analyze and compare the logs in various ways.

The paper is organized as follows. Related work is reviewed in Section 2. Section 3 describes the approach proposed. Section 4 presents a running example, which is then used in Section 5 to explain the implementation realized using YAWL and ProM. Section 6 concludes the paper by discussing the three main innovations presented in this paper.

2 Related Work

Our work combines aspects of workflow management, simulation, and process mining. Some of the most relevant contributions from these broad areas are reviewed below.

Prominent literature on workflow management [6, 13, 19] focuses on enactment, and research on workflow analysis usually focuses on verification, rather than simulation. Conversely, publications on simulation typically concentrate on statistical aspects [11, 16, 12] or on a specific simulation language [10]. Several authors have used simulation or queuing techniques to address business process redesign questions [4, 5, 14], and most mature workflow management systems provide a simulation component [7, 8]. However, none of these systems uses historic and state information to learn from the past and to enable operational decision making. We are not aware of any toolset that is able to extract the current state from an operational workflow management system and use this as the starting point for transient analysis.

In earlier work we first introduced the notion of using historic and state information to construct and calibrate simulation models [15, 20], and used Protos, ExSpect, and COSA to realize the concept of short-term simulation [15]. However, this research did not produce a practical publicly available implementation and did not use process mining techniques.

Process mining aims at the analysis of event logs [3]. It is typically used to construct a static model that is presented to the user to reflect on the process. Previously we showed that process mining can be used to generate simulation models [17], but design and state information were not used in that work.

3 Approach

A crucial element of the approach in Figure 1 is that the *design*, *historic* and *state* information provided by the workflow system are used as the basis for simulation. Table 1 describes this information in more detail.

Table 1. Process characteristics and the data sources from which they are obtained

Design information *(obtained from the workflow and organization model used to configure the workflow system)*	**Historic information** *(extracted from event logs containing information on the actual execution of cases)*	**State information** *(based on information about cases currently being enacted using the workflow system)*
• control and data flow (activities and causalities) • organizational model (roles, resources, etc.) • initial data values • roles per task	• data value range distributions • execution time distributions • case arrival rate • availability patterns of resources	• progress state of cases (state markers) • data values for running cases • busy resources • run times for cases

The design information is static, i.e., this is the specification of the process and supporting organization that is provided at design time. This information is used to create the structure of the simulation model. The historic and state information are dynamic, i.e., each event adds to the history of the process and changes the current state. Historic information is aggregated and is used to set parameters in the simulation model. For instance, the arrival rate and processing times are derived by aggregating historic data, e.g., the (weighted) average over the last 100 cases is used to fit a probability distribution. Typically, these simulation parameters are not very sensitive to individual changes. For example, the average processing time typically changes only gradually over a long period. The current state, however, is highly sensitive to change. Individual events directly influence the current state and must be directly incorporated into the initial state of the simulation. Therefore, design information can be treated as static, while historic information evolves gradually, and state information is highly dynamic.

To realize the approach illustrated in Figure 1 we need to merge design, historic and state information into a single simulation model. The design information is used to construct the structure of the simulation model. The historic information is used to set parameters of the model (e.g., fit distributions). The state information is used to initialize the simulation model. Following this, traditional simulation techniques can be used. For example, using a random generator and replication, an arbitrary number of independent simulation experiments can be conducted. Then statistical methods can be employed to estimate different performance indicators and compute confidence intervals for these estimates.

By modifying the simulation model, various 'what-if' scenarios can be investigated. For example, one can add or remove resources, skip activities, etc. and see what the effect is. Because the simulation experiments for these scenarios start from the current state of the actual system, they provide a kind of 'fast-forward button' showing what will happen in the near future, to support operational decision making. For instance, based on the predicted system behavior, a manager may decide to hire more personnel or stop accepting new cases.

Importantly, the simulations yield simulation logs in the same format as the event logs. This allows process mining techniques to be used to view the real-world processes and the simulated processes in a unified way. Moreover, both can be compared to highlight deviations, etc.

4 Running Example

Consider the credit card application process expressed as a YAWL workflow model in Figure 2. The process starts when an applicant submits an application. Upon receiving an application, a credit clerk checks whether it is complete. If not, the clerk requests additional information and waits until this information is received before proceeding. For a complete application, the clerk performs further checks to validate the applicant's income and credit history. Different checks are performed depending on whether the requested loan is large (e.g. greater than $500) or small. The validated application is then passed on to a manager to decide whether to accept or reject the application. In the case of acceptance, the applicant is notified of the decision and a credit card is produced and delivered to the applicant. For a rejected application, the applicant is notified of the decision and the process ends.

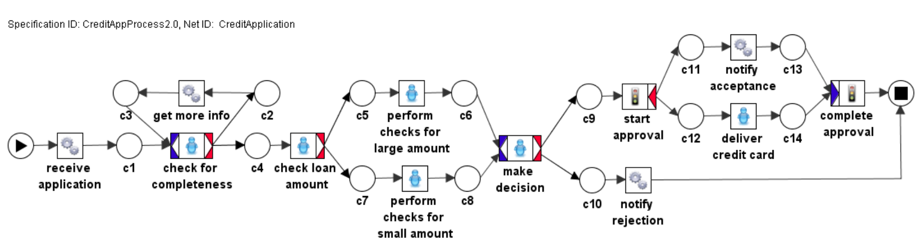


Fig. 2. A credit application process modeled in YAWL

Here we assume that this example workflow has been running for a while. In YAWL but also any other workflow system the following runtime statistics can be gathered about the long-term behavior of this process.

- Case arrival rate: 100 applications per week
- Throughput time: 4 working days on average

With respect to resources, there are eight members of staff available, which include three capable of acting as 'managers' and seven capable of acting as 'clerks'. (One person can have more than one role.)

Further assume that due to a successful Christmas promotion advertised in November, the number of credit card applications per week has temporarily doubled to 200. The promotion period is now over and we expect the rate to decrease to 100 applications per week again. However, as a result of the increased interest, the system now has a backlog of 150 applications in various stages of processing, some of which have been in the system for more than a week. Since it is essential that most applications are processed before the holiday season, which begins in a fortnight from now (the 'time horizon' of interest), management would like to perform simulation experiments from the current state ('fast forward') to determine whether or not the backlog can be cleared in time.

5 Realization through YAWL and ProM

We now use the example introduced in Section 4 to describe our proof-of-concept implementation supporting the approach depicted in Figure 1. The realization is based on the YAWL workflow environment [2] and the process mining framework ProM [1]. We focus on the new capabilities that have been added to these systems, and briefly explain the main steps that need to be performed[3].

5.1 Extracting Simulation-Relevant Information

The information contained in the workflow specification is supplemented with historical data obtained from the event logs and data from the organizational model database. This was achieved by implementing two new functions in the workflow engine to export historical data from the logs for a particular specification and to export the organizational model (i.e., information about roles and resources).

In the YAWL workflow system, event logs are created whenever an activity is enabled, started, completed or cancelled, together with the time when this event occurred and with the actor who was involved. Logs are also kept for data values that have been entered and used throughout the system. Therefore, we can retrieve historical data about process instances that have finished execution.

[3] A detailed description of how to generate a simulation model including operational decision support is provided in our technical report [18]. The example files and the ProM framework can be downloaded from http://www.processmining.org.

In this work we assume that the simulation experiments are being carried out on 'as-is' process models for which historical data is available. A function has been created which extracts the historical data for a specification from the workflow engine and exports audit trail entries in the *M*ining *XML* (MXML) log format. Some sample data for the credit application example is shown in Figure 3(a). This historical data is used for mining information about case arrival rates and distribution functions for the data values used in future simulation experiments.

```
<Process>
    <ProcessInstance id="5">
        <AuditTrailEntry>
            <Data>
                <Attribute name="loanAmt">550</Attribute>
            </Data>
            <WorkflowModelElement>
                    receive_application_3
            </WorkflowModelElement>
            <EventType>complete</EventType>
            <Timestamp>
                    2008-02-29T15:20:01.050+01:00
            </Timestamp>
            <Originator>MoeW</Originator>
        </AuditTrailEntry>
         ...
    </ProcessInstance>
         ...
</Process>
```

(a) A log entry for the completion of activity 'receive application' carried out by resource MoeW with loan amount $550

```
<OrgModel>
    <OrgEntity>
        <EntityID>1</EntityID>
        <EntityName>manager</EntityName>
        <EntityType>Role</EntityType>
    </OrgEntity>
    <OrgEntity>
        <EntityID>2</EntityID>
        <EntityName>clerk</EntityName>
        <EntityType>Role</EntityType>
    </OrgEntity>
        ...
    <Resource>
        <ResourceID>PA-529f00b8-0339</ResourceID>
        <ResourceName>JonesA</ResourceName>
        <HasEntity>2</HasEntity>
    </Resource>
        ...
</OrgModel>
```

(b) An excerpt from an organizational model with roles and resources, where resource JonesA has role 'clerk'

Fig. 3. Part of an organizational model and historical data extracted from the workflow engine

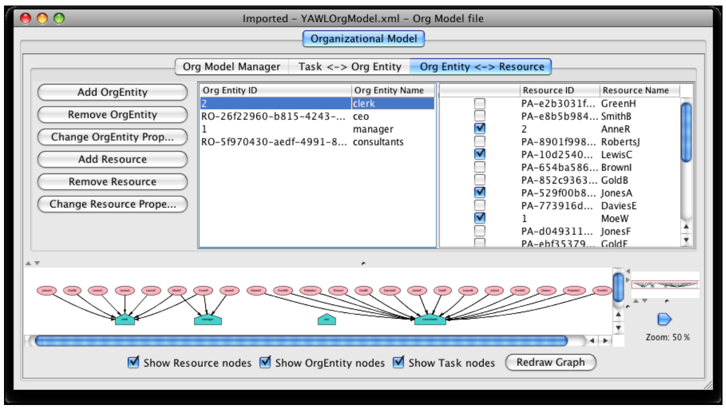
Similarly, the YAWL workflow system gives access to the organizational model through a function which extracts all available role and resource data in an organization and exports this information in the XML format required by ProM. Some sample data with the roles of clerk and manager are shown in Figure 3(b). This information is used to identify available roles and resources that are relevant for a given specification.

5.2 Generating the Simulation Model

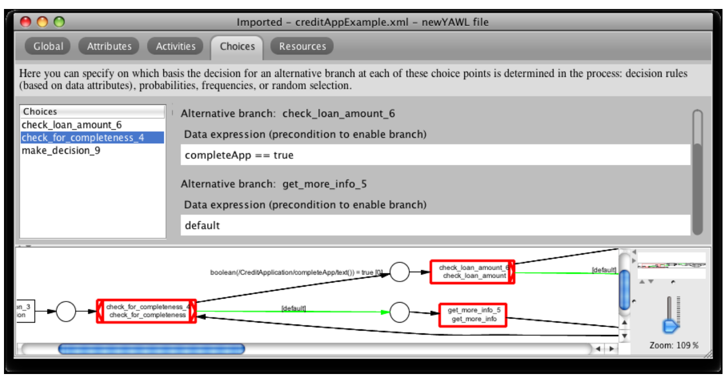
From the (1) extracted workflow specification, (2) the newly extracted organizational model, and (3) the event log file, we can now generate a simulation model that reflects the process as it is currently enacted. The direct usage of design information avoids mistakes that are likely to be introduced when models are constructed manually, and the automated extraction of data from event logs allows the calibration of the model based on actually observed parameters.

To generate the model, four basic steps need to be performed within ProM (a sample screenshot is shown for each phase in Figures 4 and 5):

1. The YAWL model, the organizational model, and the event log need to be imported from YAWL and analyzed.
2. Simulation-relevant information from the organizational model and log analysis needs to be integrated into the YAWL model.
3. The integrated YAWL model must be converted into a Petri net model (because our simulation tool is based on Coloured Petri Nets).
4. Finally, the integrated and converted model can be exported as a Coloured Petri Net (CPN) model for simulation.



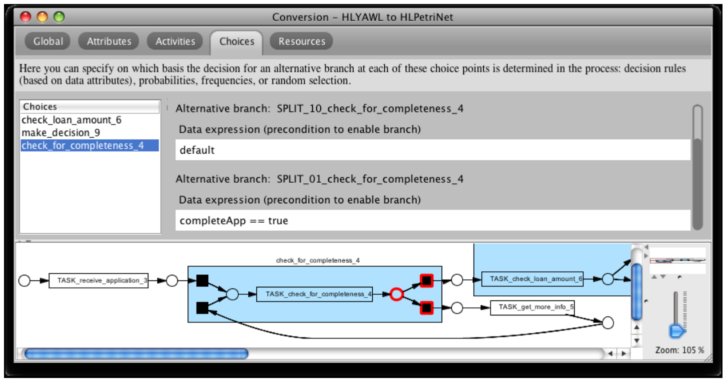
(a) Data is imported from different sources. Here the organizational model import is shown



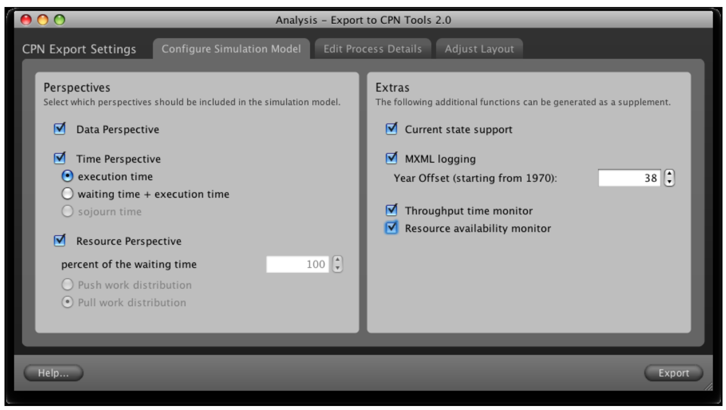
(b) The organizational model and the information obtained from the log analysis are integrated into the imported YAWL model

Fig. 4. *Phase 1*: The workflow and organizational model are imported and integrated with the information obtained from event log analysis

We can then use the CPN Tools system [9] to simulate the generated model. However, to produce useful results we do not want to start from an empty initial



(a) The integrated YAWL model is translated into a Petri net while preserving all the simulation-relevant information



(b) After importing, merging, and converting the data, a simulation model including current state support can be generated

Fig. 5. *Phase 2*: To enable the export to CPN Tools, the YAWL model is first converted into a Petri net. Then, a CPN model of the process is generated

state. Instead we load the current state of the actual YAWL system into the CPN Tools for simulation.

5.3 Loading the Current State

To carry out simulation experiments for operational decision making purposes (the 'fast forward' approach), it is essential to include the current state of the workflow system. This allows us to make use of the data values for the current cases as well as the status of the work items for current cases within the simulation experiments. A new function has been created to extract current state information of a running workflow from the YAWL system and to export this information as a CPN Tools input file (see Figure 6).

```
fun getInitialCaseData() = [(41, {loanAmt = 1500,completeApp = false,decideApp = false}),
            (40, {loanAmt = 0,completeApp = false,decideApp = false}),
            (39, {loanAmt = 500,completeApp = false,decideApp = false})];
fun getNextCaseID() = 42;
fun getInitialTokensExePlace(pname:STRING) = case pname of
            "TASK_check_for_completeness_4`E"=>[(41,"-154","JonesA")] | _ => empty;
fun getInitialTokens(pname:STRING) = case pname of
            "Process`COND_c2_15"=>[(39,"-43200")] | "Overview`Start"=>[(40,"-155")] | _ => empty;
fun getBusyResources() = ["JonesA"];
fun getCurrentTimeStamp() = "1205203218";
fun getTimeUnit() = "Sec";
```

Fig. 6. CPN Tools input file with initial state information. Several cases are in different states in the system. For example, application No. 41 is currently being checked by JonesA for completeness, and has a run time of 154 secs, i.e., ca. 2.57 mins

The following information is obtained about the current state and is introduced as the initial state of a simulation run.

- All the running cases of a given workflow and their marking.
- All the data values associated with each case.
- Information about enabled work items.
- Information about executing work items and the resources used.
- The date and time at which the current state file is generated.

When the empty initial state file of the generated simulation model is replaced with the file depicted in Figure 6, tokens are created in the CPN model that reflect the current system status (see Figure 7). For example, among the three *Case data* tokens is the data associated with application No. 41. The resource JonesA is currently performing a check activity on this case and hence, it does not appear in the list of free resources.

We now follow the scenario described in Section 4 for simulation experiments, i.e., due to a promotion 150 cases are in the system. We load the state file containing these 150 cases into the model and perform simulation experiments for the coming two weeks. We also add more resources to the model and observe how this influences the backlog and the throughput times for processing credit card applications within this time horizon.

5.4 Analyzing the Simulation Logs

We simulate the process from the generated CPN model for four different scenarios:

1. An empty initial state. ('empty' in Figure 8)
2. After loading the current state file with the 150 applications that are currently in the system and no modifications to the model, i.e., the 'as-is' situation. ('as is' in Figure 8)
3. After loading the current state file but adding four extra resources (two having the role 'manager' and three having the role 'clerk'), i.e., a possible 'to-be' situation to help clear the backlog more quickly. ('to be A' in Figure 8)

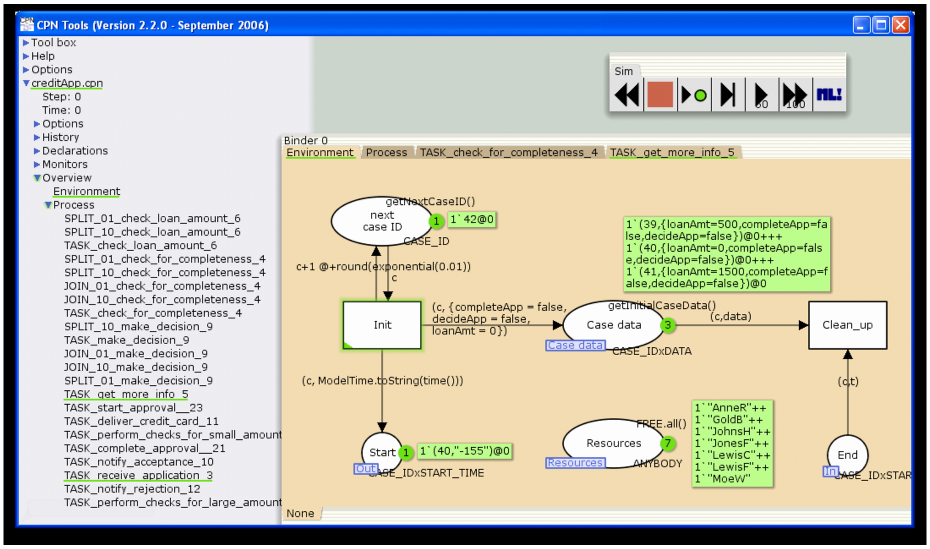


Fig. 7. The generated CPN model after loading the current state file

4. After loading the current state file and adding eight extra resources (four having the role 'manager' and six having the role 'clerk'). ('to be B' in Figure 8)

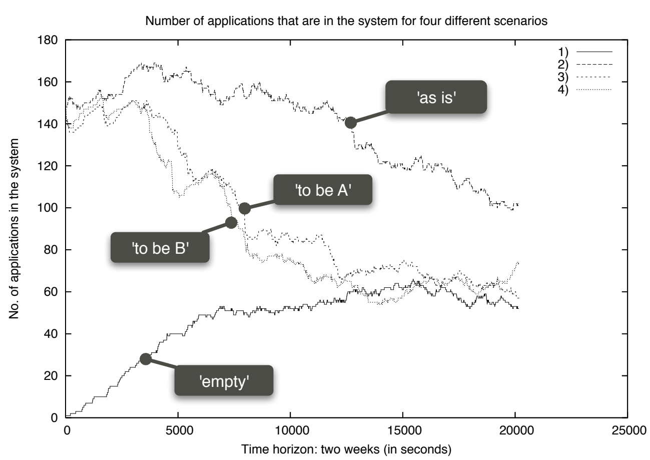


Fig. 8. Number of applications in the simulated process for the different scenarios. While the scenario with the empty state has initially 0 applications, the other scenarios are initialized by loading 150 applications from the current state file

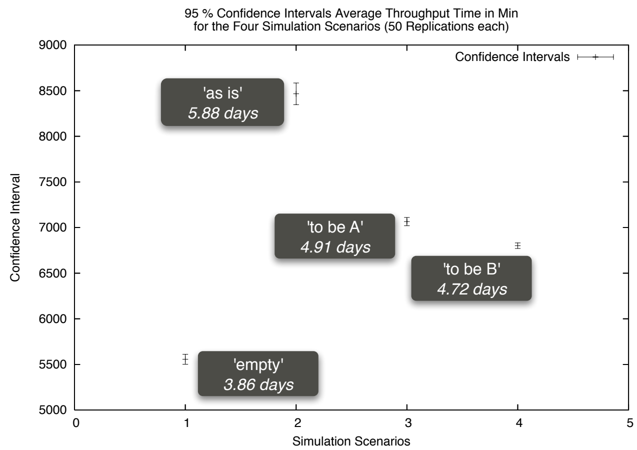


Fig. 9. Simulation run showing the 95% confidence intervals of the throughput times for the different simulation scenarios. The length of the confidence interval indicates the degree of variation

We can see the difference among these four scenarios in Figure 8, which depicts the development of the number of cases (i.e., applications) in the workflow system over the coming two weeks for an example simulation run per scenario. In the case of Scenario 1 the simulation starts with having 0 credit card applications in the system. This does neither reflect the normal situation nor does it capture our current backlog of cases. Only after a while, does this simulation represent the normal behavior of the credit card application process (i.e., with ca. 100 applications arriving per week). The other three scenarios load a defined initial state, which contains the 150 applications that we assume to be currently in the system. Furthermore, one can observe that in the scenarios where we add extra resources to the process, the case load decreases more quickly to a normal level than without further intervention. However, the scenario 'to be B' does not seem to perform much better than the scenario 'to be A' although twice as many resources have been added. This way, we can assess the effect of possible measures to address the problem at hand, i.e., we can compare different 'what-if' scenarios in terms of their estimated real effects.

CPN Tools has powerful simulation capabilities, which we can leverage. For example, it is possible to automatically replicate simulation experiments to enable statistical analyses, such as calculating confidence intervals for specific process characteristics. For instance, Figure 9 depicts the 95% confidence intervals of the average case throughput times based on 50 replicated simulations for each of the four simulation scenarios. One can observe that the estimated throughput time for the 'empty' scenario (i.e., based on the usual situation) is ca. 4 days, while the expected throughput time for the 'as is' scenario (i.e., actually expected based on the current backlog situation) is almost 6 days.

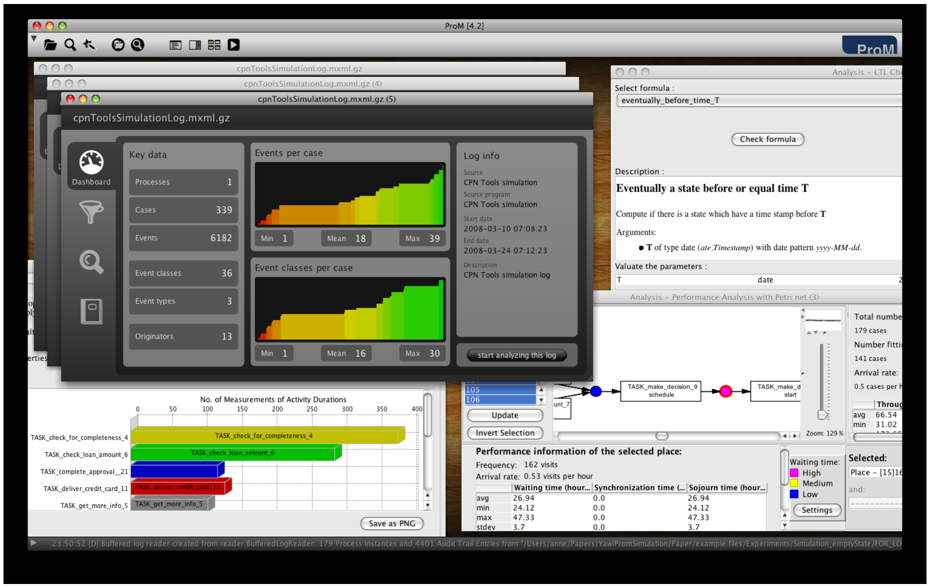


Fig. 10. The generated simulation logs can be analyzed with the same tool set as the initial workflow logs

While CPN Tools already provides powerful logging facilities and even generates gnuplot scripts that can be used to plot certain properties of the simulated process, we also generate MXML event log fragments during simulation, similar to the one shown in Figure 3(a) for the workflow log. These fragments can then be combined using the CPN Tools filter of the ProM*import* framework, which facilitates the conversion of event logs from various systems into the MXML format that is read by ProM.

The ability to use the same toolset for analyzing the simulation logs and analyzing the actual workflow logs constitutes a big advantage because the simulation analysis results can be more easily related to the initial properties of the process. In particular, since we support the loading of current cases into the initial state at the beginning of the simulation, *we can easily combine the real process execution log ('up to now') and the simulation log (which simulates the future 'from now on') and look at the process in a unified manner* (with the possibility of tracking both the history and the future of particular cases that are in the system at this point in time).

Figure 10 shows a screenshot of ProM while analyzing the simulation logs generated by CPN Tools. Various plug-ins can be used to gain more insight into the simulated process. For example, in Figure 10 the Log Dashboard (top left), the Basic Statistics plug-in (bottom left), the Performance Analysis plug-in (bottom right), and the LTL Checker (top right) are shown. The former two provide a general overview about the cases and activities in the process, whereas the Performance Analysis plug-in finds bottlenecks (e.g., in Figure 10 a bottleneck for starting the activity 'Make decision' is highlighted), and the LTL Checker can be used to verify specific properties of interest (e.g., "How many cases could be processed until they are in the stage where a decision can be made in under 3 days?").

6 Discussion

In this paper we presented an innovative way to link workflow systems, simulation, and process mining. By combining these ingredients it becomes possible to analyze and improve business processes in a consistent way. The approach is feasible, as demonstrated by our implementation using YAWL and ProM. To conclude, we would like to discuss the three main challenges that have been addressed in this research.

6.1 Faithful Simulation Models

Although the principle of simulation is easy to grasp, it takes time and expertise to build a good simulation model. In practice, simulation models are often flawed because of incorrect input data and a naïve representation of reality. In most simulation models it is assumed that resources are completely dedicated to the simulated processes and are eager to start working on newly arriving cases. In reality this is not the case and as a result the simulation model fails to capture the behavior of resources accurately. Moreover, in manually constructed models steps in the processes are often forgotten. Hence simulation models are usually too optimistic and describe a behavior quite different from reality. To compensate for this, artificial delays are added to the model to calibrate it and as a result its predictive value and trustworthiness are limited. In the context of workflow systems, this can be partly circumvented by using the workflow design (the process as it is enforced by the system) and historic data. *The approach presented in this paper allows for a direct coupling of the real process and the simulation model.* However, the generated CPN models in this paper can be improved by a better modeling of resource behavior. Moreover, the process mining techniques that extract characteristic properties of resources need to be improved to create truly faithful simulation models.

6.2 Short-term Simulation

Although most workflow management systems offer a simulation component, simulation is rarely used for operational decision making and process improvement. One of the reasons is the inability of traditional tools to capture the real process (see above). However, another, perhaps more important, reason is that existing simulation tools aim at strategic decisions. Existing simulation models start in an arbitrary initial state (without any cases in the pipeline) and then simulate the process for a long period to make statements about the steady-state behavior. However, this steady-state behavior does not exist (the environment of the process changes continuously) and is thus considered irrelevant by the manager. Moreover, the really interesting questions are related to the near future. Therefore, *the 'fast-forward button' provided by short-term simulation is a more useful option*. Because of the use of the current state and historic data, the predictions are more valuable, i.e., of higher quality and easier to interpret and apply. The approach and toolset presented in this paper allow for short-term simulation. In the current implementation the coupling between YAWL and ProM is not well-integrated, e.g., the translation of insights from simulation to concrete actions in the workflow system can be improved. Further research is needed to provide a seamless, but generic, integration.

6.3 Viewing Real and Simulated Processes in a Unified Manner

Both simulation tools and management information systems (e.g., BI tools) present information about processes. It is remarkable that, although both are typically used to analyze the same process, the results are presented in completely different ways using completely different tools. This may be explained by the fact that for a simulated process different data is available than for the real-world process. However, *the emergence of process mining techniques allows for a unification of both views*. Process mining can be used to extract much more detailed and dynamic data from processes than traditional data warehousing and business intelligence tools. Moreover, it is easy to extend simulation tools with the ability to record event data similar to the real-life process. Hence, process mining can be used to view both simulated and real processes. As a result, it is easier to both compare and to interpret 'what-if' scenarios.

Acknowledgements. This research was supported by the IOP program of the Dutch Ministry of Economic Affairs and by Australian Research Council grant DP0773012. The authors would like to especially thank Marlon Dumas, Michael Adams, Eric Verbeek, Ronny Mans, and also Christian Günther, Minseok Song, Lindsay Bradford, and Chun Ouyang plus the review team for their valuable support in implementing the approach for YAWL and ProM.

References

1. W.M.P. van der Aalst, B.F. van Dongen, C.W. Günther, R.S. Mans, A.K. Alves de Medeiros, A. Rozinat, V. Rubin, M. Song, H.M.W. Verbeek, and A.J.M.M. Weijters. ProM 4.0: Comprehensive Support for Real Process Analysis. In J. Kleijn and A. Yakovlev, editors, *Application and Theory of Petri Nets and Other Models of Concurrency (ICATPN 2007)*, volume 4546 of *Lecture Notes in Computer Science*, pages 484–494. Springer-Verlag, Berlin, 2007.
2. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
3. W.M.P. van der Aalst, H.A. Reijers, A.J.M.M. Weijters, B.F. van Dongen, A.K. Alves de Medeiros, M. Song, and H.M.W. Verbeek. Business Process Mining: An Industrial Application. *Information Systems*, 32(5):713–732, 2007.

4. R. Ardhaldjian and M. Fahner. Using simulation in the business process reengineering effort. *Industrial engineering*, pages 60–61, July 1994.
5. J.A. Buzacott. Commonalities in Reengineered Business Processes: Models and Issues. *Management Science*, 42(5):768–782, 1996.
6. M. Dumas, W.M.P. van der Aalst, and A.H.M. ter Hofstede. *Process-Aware Information Systems: Bridging People and Software through Process Technology*. Wiley & Sons, 2005.
7. C. Hall and P. Harmon. A Detailed Analysis of Enterprise Architecture, Process Modeling, and Simulation Tools. Technical report 2.0, BPTrends, September 2006.
8. M. Jansen-Vullers and M. Netjes. Business process simulation – a tool survey. In *Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools*, Aarhus, Denmark, October 2006. Published online at: http://www.daimi.au.dk/CPnets/workshop06/.
9. K. Jensen, L.M. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254, 2007.
10. D.W. Kelton, R. Sadowski, and D. Sturrock. *Simulation with Arena*. McGraw-Hill, New York, 2003.
11. J. Kleijnen and W. van Groenendaal. *Simulation: a statistical perspective*. John Wiley and Sons, New York, 1992.
12. M. Laugna and J. Marklund. *Business Process Modeling, Simulation, and Design*. Prentice Hall, Upper Saddle River, New Jersey, 2005.
13. F. Leymann and D. Roller. *Production Workflow: Concepts and Techniques*. Prentice-Hall PTR, Upper Saddle River, New Jersey, USA, 1999.
14. H. Reijers. *Design and Control of Workflow Processes: Business Process Management for the Service Industry*, volume 2617 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2003.
15. H.A. Reijers and W.M.P. van der Aalst. Short-Term Simulation: Bridging the Gap between Operational Control and Strategic Decision Making. In M.H. Hamza, editor, *Proceedings of the IASTED International Conference on Modelling and Simulation*, pages 417–421. IASTED/Acta Press, Anaheim, USA, 1999.
16. S.M. Ross. *A course in simulation*. Macmillan, New York, 1990.
17. A. Rozinat, R.S. Mans, M. Song, and W.M.P. van der Aalst. Discovering Colored Petri Nets From Event Logs. *International Journal on Software Tools for Technology Transfer*, 10(1):57–74, 2008.
18. A. Rozinat, M. Wynn, W.M.P. van der Aalst, A.H.M. ter Hofstede, and C. Fidge. Workflow Simulation for Operational Decision Support using YAWL and ProM. BPM Center Report BPM-08-04, BPMcenter.org, 2008.
19. Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer-Verlag, Berlin, Heidelberg, 2007.
20. 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. In A.H.M. ter Hofstede, B. Benatallah, and H.-Y. Paik, editors, *BPM 2007 Workshops*, volume 4928 of *Lecture Notes in Computer Science*, pages 66–77. Springer-Verlag, 2008.