Rule-based Business Process Abstraction Framework

Christina Tsagkani and Aphrodite Tsalgatidou
Dept. of Informatics & Telecommunications, National & Kapodistrian University of Athens (NKUA), Panepistimioupolis, Ilisia 157 84, Greece
{tsagkani, atsalga}@di.uoa.gr

Keywords: Business Process Modeling, Business Process Model Abstraction, BPMN 2.0.

Abstract: Business process models have become key artifacts to represent how work is performed in organisations. Therefore organisations maintain large process model repositories, containing complex models that are difficult to be managed. Abstraction is a means to reduce the size and complexity of such process models and ease the process model management. It is worthwhile noticing that several abstraction mechanisms exist in the literature but none of them includes activities-paths, data, roles, artifacts and messages in the abstraction procedure. To this end we present a conceptual framework that utilizes abstraction rules in order to simplify business process models. The proposed framework is well suited for complex BPMN process models and is built on our previous work where process abstraction rules had been introduced. This abstraction framework is designed with focus on retaining the overall structure of the process model, but most of all it is designed to treat different process elements (eg. data, messages, roles, etc.) other than activities as abstraction objects. A real business case that is originated from the financial services sector is used as an attempt to exemplify and validate the proposed mechanism.

1 INTRODUCTION

Nowadays business process management has attracted the attention by organisations and enterprises which focus on modelling and automating their business processes. Business processes can be a great source of business knowledge. Therefore, a lot of efforts are taking place by organisations in order to build process repositories containing hundreds or even thousands of business process models on behalf of different stake-holders. The way that a process model is defined is influenced by many factors, such as the modeler's expertise (e.g. software engineers, business analysts, domain experts) and the goal of modelling (e.g. models created to provide an overall view of process for managerial use, models defined in a detailed way in order to be executed by technical specialists).

After analysing diverse process models we identified that they exhibit a number of differences:

- Differences in the way that process models achieve the same effect or represent equivalent unit of work
- Differences in the granularity level used to represent the same unit of work
- Differences related to the role assigned to perform an activity
- Differences in the control-flow relations amongst activities between the process models

Both the heterogeneity that characterises business process models as it is described above and the existence of large-complex repositories, create difficulties in process model management.

Therefore we claim that such differences can be diminished and process management, especially querying, can be improved, if process models are transformed to more coarse-grained models. In this context, abstraction mechanisms are needed that take into consideration such differences and reduce the size and complexity of process models by focusing on the nature of their process elements.

Thus in this paper, we propose a conceptual framework of a rule-based abstraction mechanism that aims at providing a less complex view of business processes by focusing on significant process elements and discarding or aggregating internal process orchestration details by applying specific rules to activities-paths, data objects, messages, roles and artifacts. This mechanism focuses on business processes that are defined using the Business Process Model and Notation (BPMN) (OMG, B.P.M., 2011) as it is a well-known and widely used standard across

Tsagkani C. and Tsalgatidou A.
Rule-based Business Process Abstraction Framework.
DOI: 10.5220/0006223401730178
In *Proceedings of the Sixth International Symposium on Business Modeling and Software Design (BMSD 2016)*, pages 173-178
ISBN: 978-989-758-190-8

different industries. Therefore, the rest of the paper has been structured as follows: section 2 introduces some concepts related to the abstraction mechanism, section 3 describes the proposed abstraction framework, section 4 exemplifies and validates the proposed abstraction mechanism via a real business case, section 5 discusses related process abstraction mechanisms and section 6 concludes and presents our future work.

2 BACKGROUND

2.1 Business Process Modelling

Numerous notations have been emerged into the business process modelling space, including UML Activity Diagrams, the Business Process Modelling Notation (BPMN), Event-driven Process Chains (EPCs), Workflow nets, and the Business Process Execution Language (BPEL) that is more appropriate for executable specifications rather than modelling per se. From all these modelling notations, BPMN 2.0 is the prevailing standard, as it has been widely accepted in industrial practice and we utilise it in our abstraction mechanism.

BPMN aims at documenting and communicating business processes between all business stakeholders. Specifically, it is a graph-based notation — i.e. sets of graphical symbols and rules for combining them — for documenting flow objects, data, connecting objects, swimlanes and artifacts. Flow objects (Events, Activities and Gateways) define the behaviour of business process. Data (Objects, Inputs, Outputs and Stores) define what activities require to be performed or produce. Connecting Objects (Sequence, Message, Associations and Data Associations) define the way the flow objects are connected. Swimlanes (Pools and Lanes) represent process participants. Artifacts (Group and Text Annotation) are used to provide additional information about the process. As defined by (Chinosi and Trombetta, 2012): Process Orchestration includes the private and public processes of an organization. Private Processes are processes internal to a specific organization whereas Public processes represent the interaction between a private process and another process or participant that means only activities that are used to communicate with other participants are included in the public process. On the other hand Choreographies define the expected behaviour that is a procedural contract between interacting participants. Therefore a choreography exists between pools (or participants) as it bisects the message flows amongst them.

The current framework is focused on obtaining a process quick view by preserving the overall process structure. Therefore we suggest that the abstraction mechanism leaves intact process elements that constitute process's choreography and abstracts only process's orchestration details.

2.2 Process Abstraction

Process Abstraction is a means of providing different process views (which retain information relevant for a particular purpose) and reducing the size and complexity of process models by preserving essential properties and leaving out insignificant details (Smirnov et al., 2010). It may be applied for different purposes such as focus on specific process model properties (i.e. preserve pricey/frequent/long activities), adapt process model for an external partner, trace data/task dependencies and obtaining a process quick view respecting ordering constraints/roles.

Business process abstraction mechanisms should consider different aspects (Smirnov et al., 2012): the reason for abstraction that identifies the focus of abstraction, the conditions that should be satisfied and trigger the abstraction and the operations used for abstracting a process model to a more coarse-grained model.

Taking into consideration the above aspects the proposed abstraction mechanism focuses on providing process quick views based on defined rules that describe the conditions that should be satisfied for triggering the abstraction process and the abstraction operation (aggregation or elimination).

3 ABSTRACTION CONCEPTUAL DESIGN

In this section, we present the proposed conceptual design to process abstraction. The process abstraction mechanism is based on the use of transformation rules that are applied to business processes in a semi-automated way. The processes of the mechanism are defined with the use of the BPMN standard.

The overall process abstraction conceptual design is depicted in Figure 1 and is described in the following.

We consider that a user creates new business process models or modifies existing ones with the aid of a BPMN editor and stores them to a process repository. The process repository communicates

with the abstraction mechanism by sending existing process models and receiving the abstracted versions of the given process models.

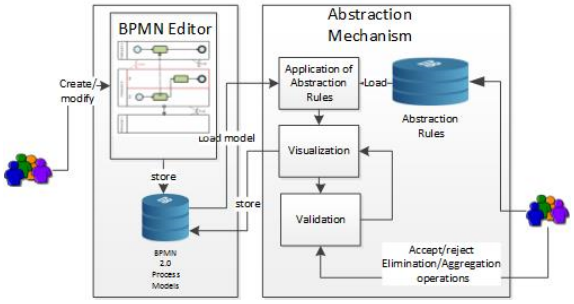

Figure 1: Conceptual design of Process Abstraction Mechanism.

The abstraction mechanism allows to apply abstraction rules to given models, visualize process elements candidate for abstraction, visualize statistics of process elements abstraction, validate abstraction and save abstraction results.

More precisely once a candidate process model of abstraction is defined in the abstraction mechanism, the mechanism communicates with the process repository and the process model is loaded into it. Then, the abstraction rules that are stored to a repository are used to trigger process abstraction. These abstraction rules presented in (Tsagkani and Tsalgatidou, 2015), are specially designed for process models defined using BPMN 2.0 and treat activities, paths, data objects, messages, lanes and artifacts as abstraction objects. These rules, when triggered, use aggregation and elimination as basic abstraction operations. Moreover, the rules are applied to the process model at hand while retaining the hierarchy in Figure 2.

The process abstraction is performed in a semi-automated way in the sense that before the finalisation of the process abstraction based on the applied rules, there is a validation phase where the user is involved. All the changes that are intended to be performed to the given process model are visualized both graphically and statistically. More precisely, each process element candidate for abstraction is clearly marked; then, it is upon the user to check its validity and accept or reject the abstraction. The resulted abstracted process model is also visualized along with some statistical results (e.g. the total reduction of activities, data, messages lanes etc.). Then, the abstracted process model is saved to the process repository where it is associated with the original business process model for future use.

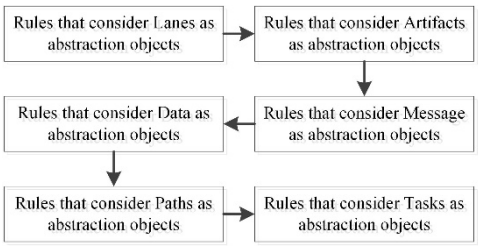

Figure 2: Abstraction Rule Hierarchy.

4 VALIDATION

In order to evaluate the proposed mechanism a detailed description of the 'Loan Approval with no Collateral' business process coming from the financial services industry was used. However in order to ensure confidentiality some details of the business process are omitted, whereas sufficient detail is provided in order to understand and illustrate the applicability of the presented conceptual framework.

Firstly, process models related to the 'Loan Approval with no Collateral' process were created based on the given detailed description, using the BPMN2Modeler that is an Eclipse-based graphical BPMN 2.0 model Editor (BPMN2Modeler, 2013). Therefore five process models were created (one model for each sub-process) – 'Requirements Capturing and Loan Quote with no Collateral via Call Center', 'Loan Request Analysis and Pre-approval', 'Final Loan Approval', 'Processing of Loan Contractual Documents' and 'Loan Disbursement'. Then the rules that are defined for the specific abstraction mechanism were manually applied to the process models in the order that is presented in Figure 2 and validated. Finally the abstracted models were produced along with some statistical data that is explained below in this section.

Due to lack of space, in this paper we use for illustration purposes a small fragment of the 'Loan Request Analysis and Pre-approval' sub-process that is shown in Figure 3. It mainly presents the tasks needed to perform loan pre-approval, the departments and the roles involved, the messages that are exchanged and the data objects and artifacts that are associated to each task. The abstracted model that is the outcome after applying the defined abstraction rules of the proposed mechanism to the model (Figure3), is presented in Figure 4.

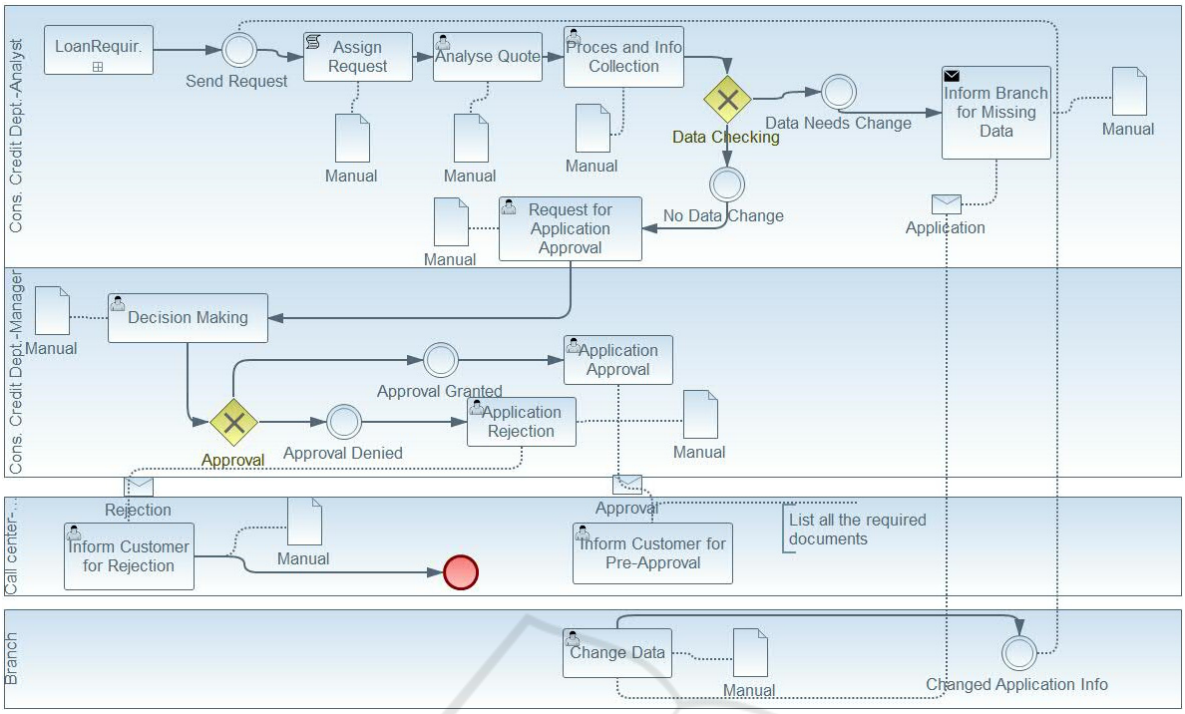

Figure 3: Fragment of the 'Loan Request Analysis and Pre-approval' process model.

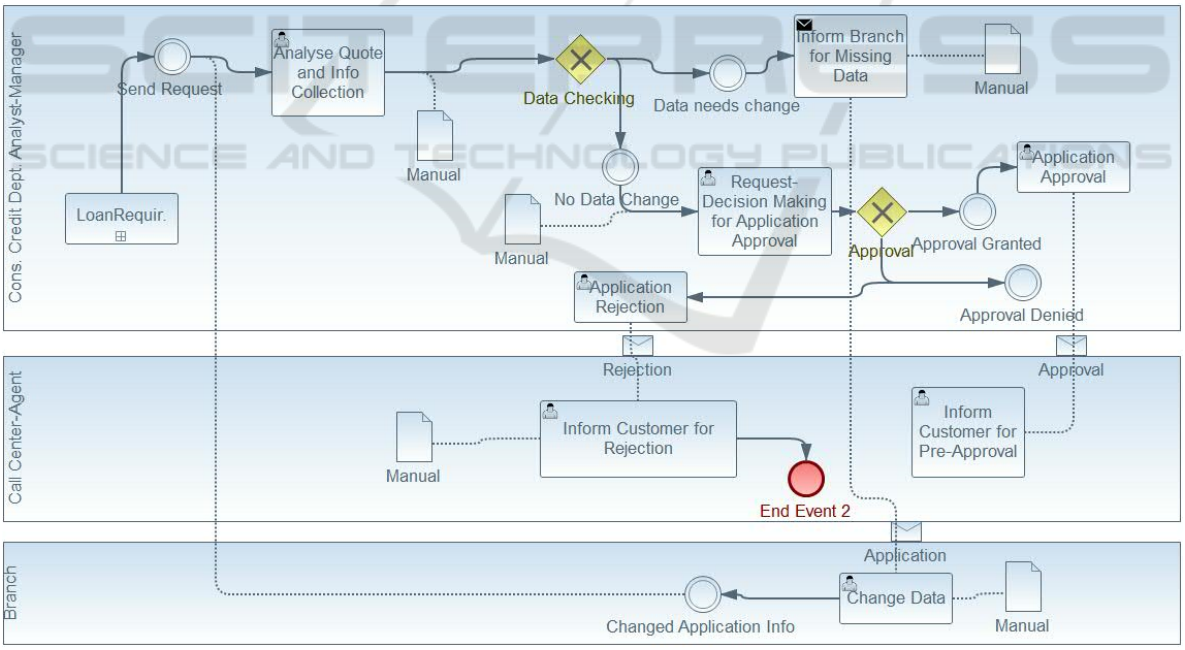

Figure 4: Abstracted Fragment of the 'Loan Request Analysis and Pre-approval' process model.

It can be observed that the two roles that co-exist in the same Consumer Credit Department are aggregated in the abstracted model. Also the script task is eliminated, the user tasks that are enclosed between the 'Send Request' event and the gateway that follows are aggregated. Moreover the user tasks that are enclosed between the 'No Data Change' event and the gateway that follows are aggregated. The same applies to their data objects that are associated with the specific user tasks. The send task

'Inform Branch for Missing data' and the user tasks 'Application rejection', 'Application Approval', Inform Customer for Pre-Approval', 'Inform Customer for Rejection' and 'Change Data' are not eliminated in the abstracted model as they are associated with message exchange and are part of the process's choreography. Finally the artifact associated with the user task 'Inform Customer for Pre-Approval' is eliminated.

The findings from applying the proposed abstraction framework to the 'Loan Approval with no Collateral' process are presented in Figure 5 where the total number of each process element that is affected by the abstraction process is presented before and after the abstraction. The findings illustrate that the activities that constituted the process were radically reduced from 70 to 43 that is 38.57% reduction. Moreover data objects show a 33.33% reduction, lanes 15.38% reduction, messages 22.22% reduction and artifacts 100% reduction.

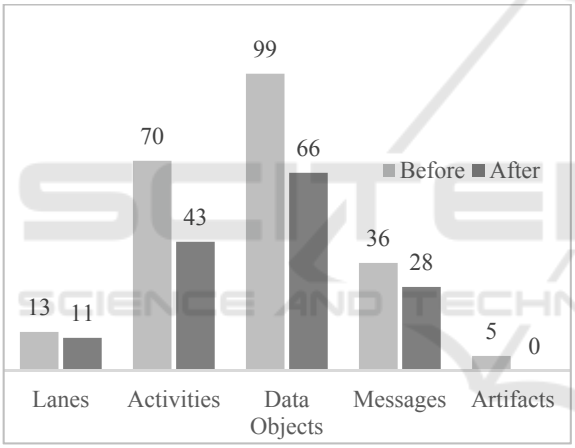

Figure 5: Total number of process elements before and after abstraction.

The business case used in this section to illustrate the value of our proposed mechanism and the findings produced seem to fulfil our research purposes that is to provide quick views of business processes while retaining the overall structure of the process and getting rid of insignificant details. Also prove that by introducing rules to the abstraction process related to process lanes, messages, artifacts and data objects (besides activities) may, to a great extent, reduce the size and complexity of the abstracted process models.

It is worthwhile mentioning that the proposed rule-based process abstraction has been also applied to other cases as well, with similar results. Some of the results have been briefly presented in (Tsagkani and Tsalgatidou, 2015).

5 RELATED WORK

There are a number research works that analyse business process models and base their abstraction mechanism based on transformation rules (Bobrik et al., 2007), (Cardoso et al., 2004), (Eshuis, and Grefen, 2008), (Günther and Van Der Aalst, 2007), (Meyer and Weske, 2012), (Pankratius and Stucky, 2005), (Polyvyanyy et al., 2008), (Sadiq and Orlowska, 2000) and (van Dongen et al., 2007). These research works not only analyse different process modelling notations and approaches but, in most of the cases, they also concentrate on activities as abstraction objects. An exception is the work by (Meyer and Weske, 2012) that considers data as abstraction objects. To the best of our knowledge the work of (Smirnov, 2009) is the only one that analyses BPMN 1.2, but it differs from our work as it suggests rules focused primarily on activities, while it only describes how other elements are influenced by activity abstraction.

The mechanism proposed in this paper targets business processes defined in BPMN 2.0 and is superior to other research works as it respects process's overall structure and proposes abstraction rules that are focused on not only activities as abstraction objects but data, messages, artifacts, paths and process participants as well.

6 CONCLUSION AND FUTURE WORK

This paper presented the conceptual framework of a rule-based process abstraction mechanism. This framework preserves the structure of the initial model, focuses on process choreography and concentrates on certain abstraction objects (participants, activities, data objects, messages and artifacts). Furthermore the proposed framework was exemplified and validated using a business case coming from the financial services industry and the findings were discussed.

We are currently working on the detailed design and implementation of a tool based on the proposed conceptual framework of the abstraction mechanism. Besides, we are planning to further validate the proposed mechanism using more business cases.

REFERENCES

Bobrik, R., Reichert, M.U. and Bauer, T., 2007. Paramete-

rizable views for process visualization.
BPMN2 Modeler Project, 2013. Eclipse.org, software available at h*ttp://projects.eclipse.org/projects/soa.bpmn2-modeler*.
Cardoso, J., Sheth, A., Miller, J., Arnold, J. and Kochut, K., 2004. Quality of service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web, 1(3), pp.281-308.*
Chinosi, M. and Trombetta, A., 2012. BPMN: An introduction to the standard. *Computer Standards & Interfaces, 34(1), pp.124-134.*
Eshuis, R. and Grefen, P., 2008. Constructing customized process views. *Data & Knowledge Engineering, 64(2), pp.419-438.*
Günther, C.W. and Van Der Aalst, W.M., 2007. Fuzzy mining–adaptive process simplification based on multi-perspective metrics. *In Business Process Management (pp. 328-343). Springer Berlin Heidelberg.*
Meyer, A. and Weske, M., 2012. Data support in process model abstraction. *In Conceptual Modeling (pp. 292-306). Springer Berlin Heidelberg.*
OMG, B.P.M., 2011. Notation (BPMN) Version 2.0 (2011). *Available on: http://www. omg. org/spec/BPMN/2.0.*
Pankratius, V. and Stucky, W., 2005, January. A formal foundation for workflow composition, workflow view definition, and workflow normalization based on petri nets. *In Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling-Volume 43 (pp. 79-88). Australian Computer Society, Inc.*
Polyvyanyy, A., Smirnov, S. and Weske, M., 2008, November. Reducing Complexity of Large EPCs. *In MobIS (pp. 195-207).*
Sadiq, W. and Orlowska, M.E., 2000. Analyzing process models using graph reduction techniques. *Information systems, 25(2), pp.117-134.*
Smirnov, S., Reijers, H.A., Weske, M. and Nugteren, T., 2012. Business process model abstraction: a definition, catalog, and survey. *Distributed and Parallel Databases, 30(1), pp.63-99.*
Smirnov, S., Weidlich, M. and Mendling, J., 2010. Business process model abstraction based on behavioral profiles. *In Service-Oriented Computing (pp. 1-16). Springer Berlin Heidelberg.*
Smirnov, S., 2009, July. Structural aspects of business process diagram abstraction. *In Commerce and Enterprise Computing, 2009. CEC'09. IEEE Conference on (pp. 375-382). IEEE.*
Tsagkani, C. and Tsalgatidou, A., 2015, October. Abstracting BPMN models. *In Proceedings of the 19th Panhellenic Conference on Informatics (pp. 243-244). ACM.*
van Dongen, B.F., Jansen-Vullers, M.H., Verbeek, H.M.W. and van der Aalst, W.M., 2007. Verification of the SAP reference models using EPC reduction, state-space analysis, and invariants. *Computers in Industry, 58(6), pp.578-601.*