Netgrif Application Engine

Gabriel Juhás[1,2], Tomáš Kováčik[1,2], Jakub Kovář[1,2], Martin Kranec[1,2] and Ľuboš Petrovič[2]

[1]*Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovakia*
[2]*NETGRIF, s.r.o., Slávičie údolie 106, 811 02 Bratislava, Slovakia*

Abstract
This paper summarizes the capabilities of the Netgrif Application Engine - a tool for interpreting Petri net based models enriched by data, roles and actions reacting to events on process and data.

Keywords
Petriflow, Petri nets, process engine

1. Introduction

The Netgrif Application Engine is the backbone of the Petriflow platform for process-driven programming. Process-driven programming is an emerging paradigm that aims to build upon the ideas of object-oriented programming (OOP), business process modeling (BPM), event-driven programming (EDP), and relational databases (RDB) in order to provide a low-code, agile platform for the design, prototyping, development, and deployment of process-driven applications [1].

There are many process execution engines available, such as Camunda or Bonita. These are BPMN-based and must therefore suffer the structural limitations imposed by the notation, such as the inability to create non-freechoice constructs [2]. During the last 30 years Petri nets have shown a continuous popularity and durability as process modelling language. While some process modelling languages were crossed with others or even disappeared, Petri nets are continuously used as modelling language addressing various purposes in the context of business processes [3]. They emerged as a solid foundation for Business Process Management research [4].

Up to our knowledge there is no tool broadly used by commercial applications, that would use Petri nets for process modelling. We believe that Petri nets deserve to have an extension, that is not only accepted by the academic community (such as various types of high-level Petri nets with tools that support them, such as CPN), but also is expressive enough and is easy-to-use,

Proceedings of the Demonstration Resources Track, Best BPM Dissertation Award, and Doctoral Consortium at BPM 2021 co-located with the 19th International Conference on Business Process Management, BPM 2021, Rome, Italy, September 6-10, 2021
gabriel.juhas@stuba.sk (G. Juhás); tomas_kovacik@stuba.sk (T. Kováčik); jakub.kovar@stuba.sk (J. Kovář); martin.kranec@stuba.sk (M. Kranec); petrovic@netgrif.com (Petrovič)
0000-0001-8302-5112 (G. Juhás); 0000-0003-3157-8609 (T. Kováčik); 0000-0002-7775-3698 (J. Kovář); 0000-0002-9432-1344 (M. Kranec)

so that it can be successfully used in commercial applications - only as much as is necessary should be added.

A short demonstration of most of the capabilities of the Petriflow platform, consisting of creating a small process-driven application, including all the steps beginning with modeling, adding data forms, introducing roles and actions, as well as deploying and demonstrating the developed application both in a standard and a public anonymous setting can be seen in our demonstration video at https://www.youtube.com/watch?v=iU1QGPUnXUs.

2. Features of the application engine

Netgrif Application Engine (NAE) is the tool for deploying and running process-driven applications written in Petriflow - a high level, Petri net based language for modeling and programming of applications.

Petriflow processes consist of a workflow model, roles, data and actions. As a workflow model, Petriflow uses place/transition Petri nets enriched by reset arcs, inhibitor arcs and read arcs.[5][6] Transitions of Petri nets represent tasks of workflow models. The Role layer defines who can execute tasks. Data variables represent all attributes of a process instance, called a *case*, during its life-cycle. Data variables associated with workflow tasks define data fields and create task forms. Data variables or places can be associated with arcs to dynamically change their weights, motivated by the concept of self modifying nets [7]. Workflow model, roles, data variables and data fields defining task forms are stored in XML format. Actions are pieces of Groovy code that define reactions to events on tasks (assign event, finish event, cancel event) and events on data fields. Actions can trigger events and call external functions.

NAE is composed of several building blocks. The core of the NAE is the Process Engine Server, that enables us to:

- Upload, run and delete Petriflow processes
- Create, run and delete instances of processes
- Assign, finish and cancel tasks of process instances
- Read and save data variables
- Execute actions

This core is extended with various modules that enhance its functionality in diverse ways.

The Rule Engine is the first of these extensions. The motivation for the creation of this module was, to add or remove rules to the Rule Engine without the need to deploy updated versions of the affected processes. Another big advantage of the Rule Engine is that sometimes there are business requirements that span across multiple different processes and you want to satisfy them in a centralised way. The Rule Engine allows you to specify if-then decision rules written in MVFLEX expression language (MVEL). Whenever an event in the Process Engine Server is triggered, it checks the Rule Engine for any rules with satisfied conditions for the event. In such a case, the Rule Engine executes the then statement of the rule. The statements are Petriflow actions and can trigger additional events in the Process Engine Server. The Rule Engine is based on Drools.

The second extension is the Process Engine Client which provides an Angular based user interface to the Process Engine Server. The client communicates with the server via a REST API. The process engine client consists of a library of Angular components, that provide integration with most of the capabilities of the Process Engine Server. The most important capabilities include the task list, that displays the available tasks to the user and allows them to interact with them by changing their state, filling their data and applying validations to user input. Another very important capability is the public view, that allows anonymous users to create instances of public processes and to execute tasks of these instances. The Petriflow processes can be designed in such a way that only a part of the model is publicly available and the data filled by the anonymous users can then be processed by an internal staff.

The given extension can be used with an already pre-made design that generates customisable views using configuration files, alternatively you can define your own library of components with your own design, which you can use in place of the Process engine client. Since the Process Engine Server has a published REST API, you do not need to restrict yourself to using our implementation of the Process Engine Client. The Process Engine Server can be used on its own and can be integrated with any technology and environment. This architecture allows the users of our tools to choose what is best for their needs, whether it is just an environment for running pure Petriflow based application, or running them in a customised environment with special, custom made, features, or using the Process Engine Server as an addition to an already established software environment.

The Process Engine Server, the Rule Engine, the Process Engine Client together with a set of predefined processes for managing roles, users, authentication and authorization of users and deploying of Petriflow processes is provided as a Default Process Driven Application. The Default Process Driven Application allows administration of processes by deploying and deleting the processes, invitation of users and management of user roles. It offers the functionality of registering users. For registered users, it offers the functionality of the task list, where they can assign and perform the tasks associated with one of their roles. For anonymous users, it offers the ability to fill public task forms of public processes and to access the instances of the public processes via the ID of the process instance.

A demo of the Netgrif Process Driven Application, consisting of the Process Engine Server together with the predefined Process Engine Client, where you can deploy your Petriflow processes and execute the process instances, is available free of charge for noncommercial use at https://demo.netgrif.com/.

Introduction materials, how-to articles and tutorials explaining the principles of the Petriflow language, as well as modelling and development guides can be found at https://netgrif.com/started/.

3. Use cases of the Petriflow platform

The main use case of the low code Petriflow platform is its capability of creating rich process based, easily configurable applications. An advantage and use case of the platform in an academic setting, is the ease of illustrating the versatility of Petri nets as a platform for business process modelling and process based application development, without the need to introduce additional modelling formalisms to the students, such as BPMN.

The programming language Petriflow, as well as our ideals and techniques of process-driven programming are being thought as part of a masters degree curriculum at the Slovak University of Technology in Bratislava. During the first half of the course the students are introduced to the Petriflow language and the concepts of process-driven programming. The course builds on top of previous courses that have introduced Petri nets, modelling, object-oriented programming and various server and client side frameworks, that help them get deeper understanding of the entire application engine stack [8].

The second half of the course focuses on group work and allows the students to go through the whole life cycle of the digital transformation for their chosen use case. They create process models, add roles and associate them with the tasks, add data, add actions for computation and inter-process communication between process instances, and finally deploy the processes and create filters of the cases and/or task as menu items, resulting in the final application. The groups of students are capable of creating complex applications in just a few weeks, which highlights the advantages of the Petriflow platform for rapid development and its huge potential for business application [8]. Since 2018, when the Petriflow language was introduced into the course, over 350 students have participated in the course and have created more than 60 process-driven applications.

The best students are offered participation via student jobs in real projects of digital transformation for real customers, where they help the experienced professionals to deliver Petriflow based applications. The students also regularly participate in meetings with business users of the developed applications [8]. About 30 students have been a part of this initiative, many of which have been offered full-time positions after graduating from the university.

In total, 10 different customers participated in the project as partners, including insurance companies, leasing companies, an electricity distribution company, a media broadcasting company and a healthcare provider.

During the second wave of the COVID-19 pandemic the Slovak government introduced weekly antigen testing to fight the spread of the virus. The capacity for testing was provided by mobile testing sites. An application was needed for the management of the testing sites, medical teams, scheduling of testing days and reporting of the results to the National Center for Health Information. An application that provides this functionality called Arthur has been developed and deployed into production with the help of the Petriflow platform in just 12 days.

All these projects were implemented using the Petriflow language and the Netgrif Application Builder and run in the Netgrif Application Engine in production [8], processing over 100 million instances of the various business process. NAE is successfully used by Netgrif partners to develop various other applications.

More information about the use cases of the tool can be found online at https://netgrif.com/case-studies/.

4. Conclusion

In this paper, we presented the Netgrif Application Engine a platform, that enables the creation of rich process based customisable applications with the use of the low-code language Petriflow.

The platform aims to be an ambassador for the use of Petri net based modelling languages not only in academical, but also in commercial setting.

References

[1] G. Juhás, Process-driven programming, 2021. URL: https://netgrif.com/process-driven-programming/.

[2] W. van der Aalst, On the representational bias in process mining, in: 2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE, 2011. URL: https://doi.org/10.1109/wetice.2011.64. doi:10.1109/wetice.2011.64.

[3] A. Koschmider, A. Oberweis, W. Stucky, A petri net-based view on the business process lifecycle, Enterprise Modelling and Information Systems Architectures (2018) Vol 13 (2018). URL: https://www.emisa-journal.org/emisa/article/view/192. doi:10.18417/EMISA.SI.HCM.4.

[4] W. M. P. van der Aalst, Business process management as the "killer app" for petri nets, Software & Systems Modeling 14 (2014) 685–691. URL: https://doi.org/10.1007/s10270-014-0424-2. doi:10.1007/s10270-014-0424-2.

[5] R. Lorenz, J. Desel, G. Juhás, Models from scenarios, in: Transactions on Petri Nets and Other Models of Concurrency VII, Springer Berlin Heidelberg, 2013, pp. 314–371. URL: https://doi.org/10.1007/978-3-642-38143-0_9. doi:10.1007/978-3-642-38143-0_9.

[6] G. Juhás, R. Lorenz, S. Mauser, Complete process semantics for inhibitor nets, in: Petri Nets and Other Models of Concurrency – ICATPN 2007, Springer Berlin Heidelberg, 2007, pp. 184–203. URL: https://doi.org/10.1007/978-3-540-73094-1_13. doi:10.1007/978-3-540-73094-1_13.

[7] R. Valk, Self-modifying nets, a natural extension of petri nets, in: Automata, Languages and Programming, Springer Berlin Heidelberg, 1978, pp. 464–476. URL: https://doi.org/10.1007/3-540-08860-1_35. doi:10.1007/3-540-08860-1_35.

[8] A. Juhásová, G. Juhás, L. Molnár, M. Ondrišová, J. Mažári, M. Mladoniczky, IT induced innovations: Digital transformation and process automation, in: 2019 17th International Conference on Emerging eLearning Technologies and Applications (ICETA), IEEE, 2019, pp. 322–329.