

# Netgrif Workflow Management System based on Petriflow language

Juraj Mažári<sup>1,2</sup>, Gabriel Juhás<sup>1,2,3</sup>, Milan Mladoniczky<sup>1,2</sup>, Tomáš Gažo<sup>2</sup>, and Martin Makáň<sup>2</sup>

<sup>1</sup> Faculty of Electrical Engineering and Information Technology STU in Bratislava, Ilkovičova 3, 812 19 Bratislava, Slovakia,

<sup>2</sup> NETGRIF, s.r.o.,

Blumentálska 12, 811 07 Bratislava, Slovakia

[netgrif@netgrif.com](mailto:netgrif@netgrif.com)

WWW home page: <http://www.netgrif.com>

<sup>3</sup> BIREGAL s. r. o.,

Klincova 37/B, 821 08 Bratislava, Slovakia

[biregal@biregal.sk](mailto:biregal@biregal.sk)

WWW home page: <http://www.biregal.com>

**Abstract.** Netgrif Workflow Management System (Netgrif WMS) is JVM based application build on Spring framework. Netgrif WMS can be used to import Petriflow workflows in form of xml file. Process roles are then assigned to other users by authorized user. Users can create new cases (instances) of imported workflows. New tasks are automatically generated for each new case. Users with corresponding roles are able to assign, reassign, finish or cancel task. Tasks data fields can be edited by assigned user and are automatically validated and saved.

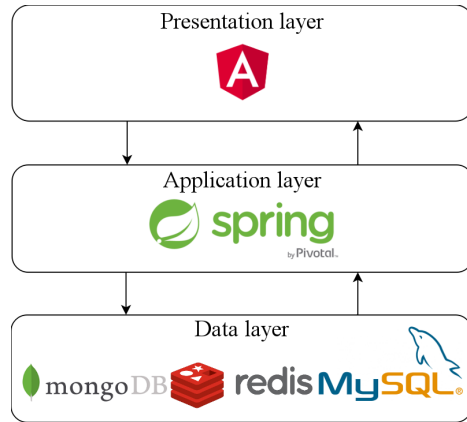
**Keywords:** Petri nets, workflow, Spring Boot

## 1 Introduction

Modern companies need workflow management systems that are able to frequently change their workflows according to changes in bussiness sphere. Petriflow language aims to provide modeling capabilities to satisfy this needs. Workflow Management System is build as a tool for simulation of Petriflow based workflows. Main application of WMS is to import Petriflow workflows and create new cases of them. Assigning, finishing and canceling task is available for registered users with correct process roles. User management is available for users with admin system role.

## 2 Architecture

WMS is build as a three-layer client-server application. AngularJS framework is used on presentation layer, Spring Boot framework on application layer and multiple databases on data layer. Figure 1 depicts three-layer architecture of Netgrif WMS.

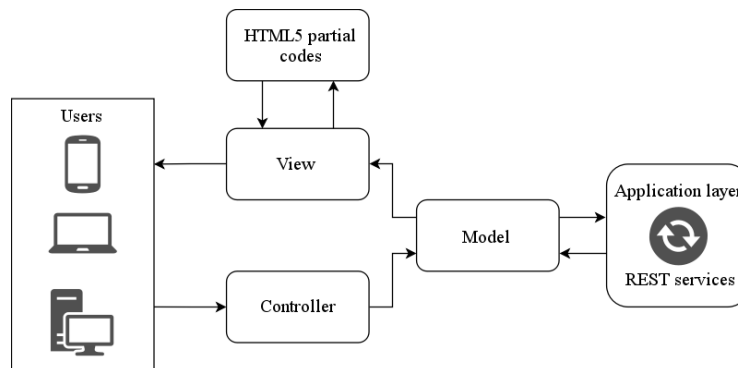


**Fig. 1.** Netgrif WMS three-layer architecture

## 2.1 Presentation layer

AngularJS framework is base stone of presentation layer. AngularJS uses the MVC design pattern. The model, view and controller are well defined in AngularJS and serve to simplify the development process. Figure 2 shows communication between those components, web browsers and application layer [1].

Developers can easily create AngularJS service that communicates with multiple web-services. Services can be used on many places, this reduce number of code lines. Along with AngularJS, Netgrif WMS also uses AngularJS Material



**Fig. 2.** AngularJS internal and external communication

framework. It provides a set of reusable UI components based on Google's Material Design. This gives Netgrif WMS modern look on which many users are accustomed. Therefore navigation and UI components are user friendly.

## 2.2 Application layer

Application layer of WMS is build on **Spring Boot** framework. In accordance with rapid development, Spring Boot makes it easy to create stand-alone applications that can be started by simple `java -jar` command thanks to embedded Tomcat[2]. Spring framework also introduces **dependecy injection**, which enables autowiring of components.

Application layer is separated into multiple modules separating their concerns. Each module consists of three packages - domain, service and web. Domain package contains entity classes which are persisted into database. Bussines logic is kept in service packages. According to Spring customs, interface is created and used to autowire each service. Web package keeps request and response body classes and REST controllers, which calls methods on autowired services.

## 2.3 Data layer

Data layer leverages benefits of different database systems. Structured data which are rarely changed, such as user credentials, are stored in **MySQL** database. User sessions are stored in in-memory database **Redis**. Tasks, cases, imported workflows and other unstructured and constantly changing data are stored in **MongoDB** database.

Each of these databases is easily accessible via Spring Data project. It enables to use databases without any configuration. Spring Data also reduce the amount of boilerplate code required to implement data access layers for various persistence stores.

## 2.4 Presentation-application layer communication

Presentation and application layer are connected via REST web-services. **HATEOAS** constraints are applied on those web-services. Main advantage of HATEOAS principle is that presentation layer does not have to know all web-services URLs. HATEOAS response provides data and all accessible web-services for the given resource.

## 3 Implementation

Netgrif WMS is implemented as a Spring Boot project build by **Maven**. Open source automation server **Jenkins** is used for building and deploying. This tool automatically builds and deploys new version of Netgrif WMS after each commit to git repository. This way is secured that latest version is always tested and deployed if all test passed.

Classes extending Spring Boot's **CommandLineRunner** interface are runned at startup. This is used to create dummy data for development. Sample user accounts and cases are easily generated this way. Using spring boot property

`spring.jpa.hibernate.ddl-auto` set to `create-drop` for development use allows to automatically create new version of data at startup and delete existing data on application end.

Import of workflow net has gone through many changes. Best option comes to be use of **JAXB API**. Java classes are generated from Petriflow Xsd definition. Those classes are used to unmarshall Xml file. Unmarshalled file is processed by importer and persisted into MongoDB database as domain objects which are used by Netgrif WMS. Original Xml file is also saved to local directory. Later it can be used to get a snapshot of state of any case in the system. This is very useful for debugging during development phase. It is also possible to see a complete list of all data fields and their values for any case. These features are only available for development environment.

Generated tasks can be assigned by user himself or delegated by another user. This allows both pull and push control[3]. Both actions are possible only for users with correct process roles.

Netgrif WMS supports three types of triggers: user, automatic and time trigger. Tasks with automatic trigger are finished by system immediately. Time triggers can be set to specific date and time or specific delay from current time.

## 4 User interface

Users can be invited by admin users through **admin console**. Admin can define users email, organizations and process roles for imported processes. This will send email invite to specified address. Invite contains link to registration form in which user have to enter his real name and password. Users email is used as login. In the next tab, admin can manage process roles of already registered users.

Netgrif WMS provides two different views for managing tasks. **Task view** displays all available and assigned tasks for logged in user. Task panel is expanded upon clicking on it. Expanded task panel shows all visible data fields. User can change each editable data field and new value is automatically validated and saved. Invalid values are highlighted and task containing invalid data fields can not be finished.

Second view is named **case view**. Case view displays all available cases. After clicking on case panel a new tab is opened. Tasks belonging to selected case are displayed in that tab. Each task panel behaves the same way as in task view. This enables users to work with multiple cases simultaneously. User can create new case by clicking on a **+** button in tabs panel. This opens a dialog which enables user to select desired workflow, case title and label color.

## 5 Conclusion

In this paper we introduced Netgrif Workflow Management System application, which is based on modern principles and technologies. It leverages main advan-

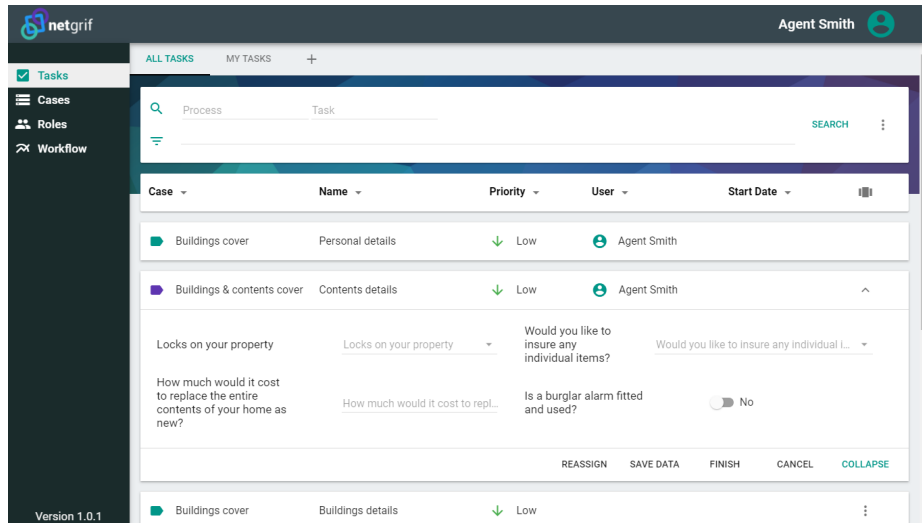


Fig. 3. Example of task view in Netgrif WMS

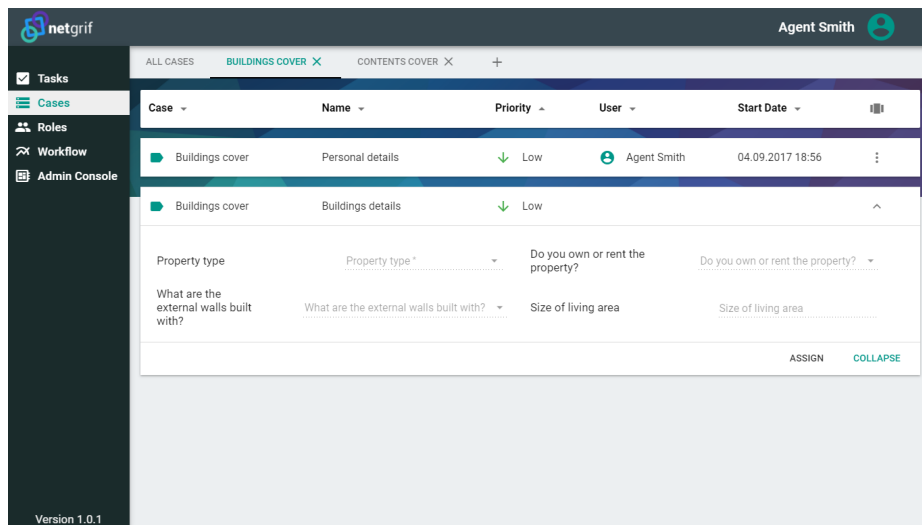


Fig. 4. Example of case view in Netgrif WMS

tages of Petriflow language and supports rapid deployment of frequently changing workflows. From real world experiences we have learned this feature is highly valued and required by many companies.

Future development includes new architecture. With Spring Boot 2.0 coming in December 2017 we plan to go **reactive** with Netgrif WMS. We are going to break the whole project into separate and independent **microservices**. Spring Boot 2.0 will introduce support for reactive programming. This will give us non-blocking, event-driven and easily scalable application. Presentation layer will also undergo changes. We are going to create UI service that will put together UI client from UI components based on workflow definition. We plan to use event sourcing and Command Query Responsibility Segregation for some microservices. This will provide us with a complete history of events and changes in any case.

## References

1. Williamson, K.: Learning AngularJS: A Guide to AngularJS Development O'Reilly Media, Inc., (2015)
2. Walls, C.: Spring Boot in action Manning Publications Co., (2016)
3. Van der Aalst, W. M.: The application of Petri nets to workflow management. Journal of circuits, systems, and computers, 8.01, 21-66 (1998)