I.J. Intelligent Systems and Applications, **2022, 3, 54-73**
Published Online June 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2022.03.05

Towards an Intelligent Approach to Workflow Integration in a Quality Management System

Mohamed Nazih OMRI
MARS Research Laboratory LR17ES05, University of Sousse, Tunisia
E-mail: mohamednazih.omri@eniso.u-sousse.tn

Hadhemi BEN AONNE
MARS Research Laboratory LR17ES05, University of Sousse, Tunisia
E-mail: benounnehad@gmail.com

Received: 02 January 2022; Revised: 11 February 2022; Accepted: 15 March 2022; Published: 08 June 2022

Abstract: Among the most important activities within a company, we find that of quality management. This activity represents reflects the most rigorous way possible for a better organization of establishments in order to offer the best service to customers and to the various members of these establishments. This activity of quality management is a very delicate and sensitive task due to the large number of documents and business processes that are handled on a cyclical basis. For this reason, setting up a reliable and efficient system for managing the different aspects of the quality management process becomes a challenge for any company that seeks excellence. This article proposes a new intelligent approach to the need of the management of human and commercial resources within the companies for a good management of the process of quality management according to its own conception. Our approach allows any quality management manager to manage the different modules of a QMS according to the ISO 9001 standard through the different interfaces offered by our solution. The monitoring phase of this process through the implementation of a workflow orchestrator, jBpm.

Index Terms: QMS, quality management, process, workflow, jBPM, orchestrator.

1. Introduction

1.1. Context and issue

Traditionally, the activity of establishments is counted and recorded on paper, then in Excels binders and organized directories. These tools are not adequate to allow an evaluation of the performance of the collaborators. IT companies represent a booming sector and have been facing, for several decades, a change imposed by their supervisors and their users. It is essential that the services offered there are of good quality to best meet customer demand. In fact, there is some customer dissatisfaction due, generally, to the wait times that are estimated to be too long or to the mismatch between their needs and the products offered. These establishments therefore have a strong need to implement more efficient and rigorous management of resources, with the aim of improving the productivity and efficiency of their system while ensuring the improvement of the quality of products supplied to them clients.

1.2. Goal

Among the most important activities for companies, we find the activity of quality management. The latter represents a rigorous way to properly organize establishments in order to produce good service to customers as well as to the various members of the establishment. This is a very sensitive task because of the large number of documents and business processes processed cyclically. This requires a reliable and efficient system to properly manage the different aspects of a quality management system. The workflow integration approach in a quality management system as part of this master's degree is based on an architecture, which allows through three phases (Modeling, generation, execution) to put an operational and autonomous workflow. The main objective of this approach is to automate the various quality processes according to the international standard ISO 9001 through the interaction of the various actors involved in the processes. In this paper, we propose an Intelligent new Approach to Workflow Integration in a Quality Management System.

1.3. Contributions

We propose a new approach for integrating workflow into a quality management system. We begin by presenting the general architecture of the workflow system as a whole. We then present our integration solution between the jBPM workflow engine, and the various processes developed and implemented. And, finally, we detail the principle of communication with the Spring framework and the Drools rule engine before presenting the different interfaces and the implementation of the different modules of the proposed system.

1.4. Paper structure

The rest of this article is structured as follows. In the section 1, we discuss the state of the art on quality management systems, the ISO 9001 standard as well as the various solutions existing in the market. We study the main approches presented in the literature. In the section 2, we present the notion of workflow engine with the different functionalities. Section 3 is dedicated to the presentation of the integrity approach between a quality management system and a workflow engine. We will also develop the notion of a rule engine and the integrity between jBPM, our workflow engine, and the Spring framework. In section 4, we present and detailed the different tools used in our approach. Section 5 presents the experiments carried out on our approach and the analysis of the results. In section 6 we give a discussion and a certain limits of the presented work. We conclude this article with a general conclusion and some prospects in the last section 7.

2. Related Work

To properly develop and implement an idea and propose it as a new solution, it is imperative to start by highlighting the state of the art to highlight the limits and shortcomings of existing solutions. In this section we will define the quality management system, the different functions, as well as the objective of a QMS.

2.1. Quality management system

A Quality Management System is the set of actions implemented by a company that wishes to have a quality or continuous improvement approach in order to increase the quality of its production and its organization and to obtain consistent results and predictable effectively [1]. Figure 1 illustrates the evolution of the quality approach over time.

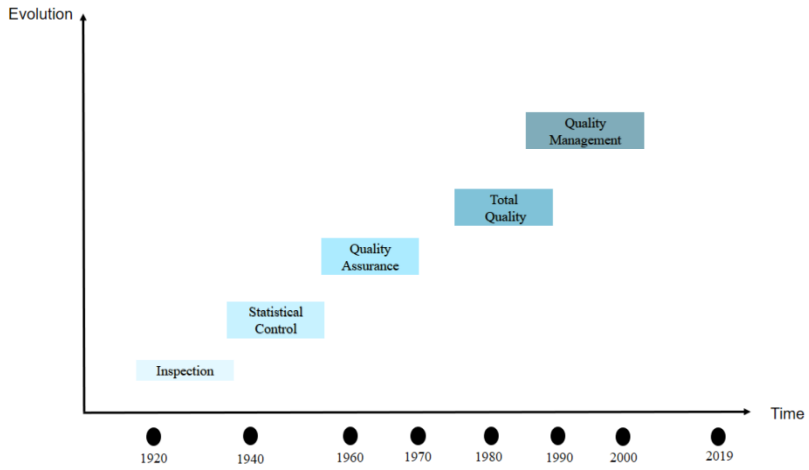

Fig.1. Evolution of the concept of quality [2].

Like all techniques that have experienced significant development during this century, concepts of quality have continued to evolve from the statistical control of finished products invented in the United States in the 1920s and widely applied during WWII to the concepts of total quality management considered to be the challenge of the 90s [2]. The evolution of quality can be summed up in three ages:

- Sorting age (from the 40s until the 60s) when manufacturers are aware of the need to control finished products to guarantee a certain quality, each part is controlled and discarded if it is defective, which was extremely costly for the company.
- Age of control (from the 60s until the 80s) during production on certain characteristics of the product in order to allow corrective actions to be taken as soon as deviations from the objectives are detected.
- Age of continuous quality (From the 80s until now) improvement which consists of starting from a strategic vision on the future of the company to organize itself to be able to guarantee its customers the prescribed

level of quality and to present processes that allow" explain the different methods applied to achieve this.

The ISO 9000 family of standards corresponds to a set of repositories of good management practices in terms of quality, carried by the international standardization body (ISO, International Organization for Standardization). The ISO 9000 standards were originally written in 1987, then they were revised in 1994 and again in 2000. Thus, the ISO 9001 version 2000 standard, part of the ISO 9000 family, is written ISO 9001: 2000. The ISO 9001: 2000 standard is essentially about the processes allowing to realize a service or a product whereas the ISO 9001: 1994 standard was essentially centered on the product itself. Here is a summary presentation of the different standards of the ISO 9000 family:

- ISO 9000: "Quality management systems - Fundamentals and vocabulary". The ISO 9000 standard describes the principles of a quality management system and defines its terminology.
- ISO 9001: "Quality management systems - Requirements". The ISO 9001 standard describes the requirements for a quality management system for use either internally, or for contractual or certification purposes. It is therefore a set of obligations that the company must follow.
- ISO 9004: "Quality management systems - Guidelines for improving performance". This standard, intended for internal use and not for contractual purposes, relates in particular to the continuous improvement of performance.
- ISO 10011: "Guidelines for auditing quality management systems and/or environmental management".

2.2. SMQ according to standard ISO 9001

A Quality Management System, often abbreviated QMS (in English: Quality Management System) is the set of activities by which the organization defines, implements and reviews its quality policy and objectives in accordance with its strategy. The QMS of an organization is made up of interrelated and interactive processes using resources to achieve the targeted results and provide value (product, service, etc.) [3]. The quality management is based on seven principles according to ISO 9000: 2015:

- Customer focus.
- Leadership.
- Staff involvement.
- Process approach.
- Continuous improvement.
- Evidence-based decision making.
- Management of relations with interested parties.

According to ISO 9001, QMS is based on a balance between three criteria, the absence of one is sufficient for the failure of a project. These three criteria are:

- Time limit.
- Cost.
- Quality.

According to the report of the Project Management Institute in 2017: 38% of projects meet business goals and intentions, and 33% of projects are considered failures, which means that failures are mainly related to the project quality management (PMI, 2017).

2.3. Functions, Goals and Benefits of a QMS

The QMS is necessary for the control and the improvement of the various processes of an organization, thus allowing the continuous improvement of its results and its performances. The implementation of the quality management system includes:

- A system that documents practices (business processes, business procedures, operating methods, etc.)
- A verification system (internal audits, for example)
- A management level results analysis system (management review).

Indeed, the QMS is part of the set of management tools made available to a manager who must coordinate activities to guide and control the entity he manages (a company, an association, a service, a contract ...). The implementation of quality management therefore goes well beyond simple compliance control, a posteriori, since it involves:

- Plan: Establish the objectives and processes necessary to deliver results that match customer requirements

and entity policies.

- Develop: implement everything that must be done to guarantee, a priori, the satisfaction of the client's requirements and the satisfaction of the entity's needs.
- Check: monitor and evaluate the results obtained and determine the actions to be taken to correct the observed deviations.
- Adjust: take actions to correct deviations as well as continuously improve the performance of activities or products.

It is what we call the PDCA cycle which transforms an idea into action and action into knowledge. This cycle is presented by the Deming wheel or in English "Deming wheel", a graphic transposition of the PDCA quality management method. The figure 2 illustrates the graphical representation of the Deming wheel.

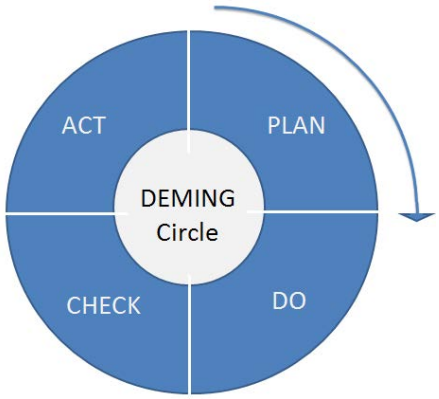

Fig.2. Deming wheel [3].

This standard is based on a number of quality management principles, including strong customer focus, management motivation and commitment, process approach and continuous improvement. ISO 9001: 2015 helps ensure that customers get consistent, high-quality products and services, with great business benefits in return. The purpose of the QMS is twofold, it makes it possible to:

- Guarantee product quality assurance.
- Increase customer satisfaction.

Successful use of the QMS with their principles can provide benefits to interested parties, such as:

- Greater financial returns,
- The creation of value,
- Greater stability.

2.4. Examples of QMS systems

There are several software programs on the market that facilitate quality management for companies, in this part we have studied the characteristics and functionalities of certain applications available while focusing on their weak points [4-10].

- Divalto Infinity Quality: Full Saas software "Software As A Service" offered by the French company Divalto Groupe, its functionality is intended for all companies certified ISO 9001 or being in a quality assurance process. It ensures them the management of their quality system as well as the following functionalities:

  - ✓ Follow-up of quality events (Complaints, Audits ...).
  - ✓ – Management of quality problems (external and internal NC, Improvement plans, Risk findings, etc.).
  - ✓ Management of immediate, corrective and preventive (Planning by manager, process, etc.).
  - ✓ Supplier follow-up (Skills, Evaluations, Qualification by service, Quality reception follow-up, Selection / Classification).
  - ✓ A measure of the costs of obtaining quality

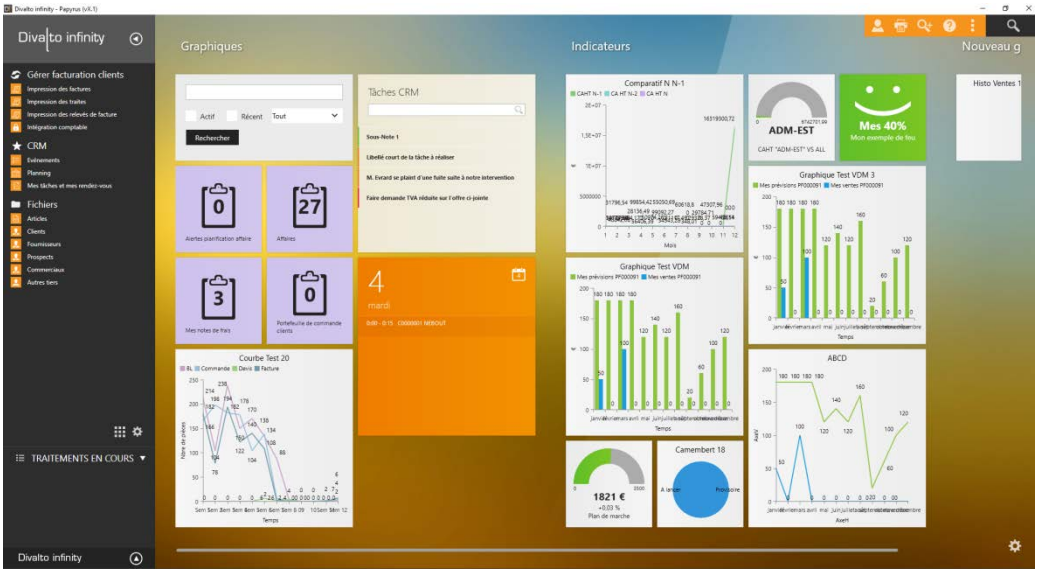

Fig.3. Divalto Infinity Quality application interface [11].

- QualiShare Quality: Qualishare Quality software is a simple and complete solution for effectively piloting and managing a quality management system. The software was designed to develop the set of improvement approaches used by the company. It allows you to manage documents, non-conformities, audits, improvement actions, performance indicators, etc. "Qualishare Qualité" has been designed and developed to meet the requirements of ISO standards (9001, 14001, 18001, and other sectoral standards: 13485, 17025, 17020, 15189, 9100, etc.) and to save time for the monitoring and improvement of the management system. It is also an essential tool for involving each employee in the process. The figure 4 shows an interface of the "QualiShare Quality" application.

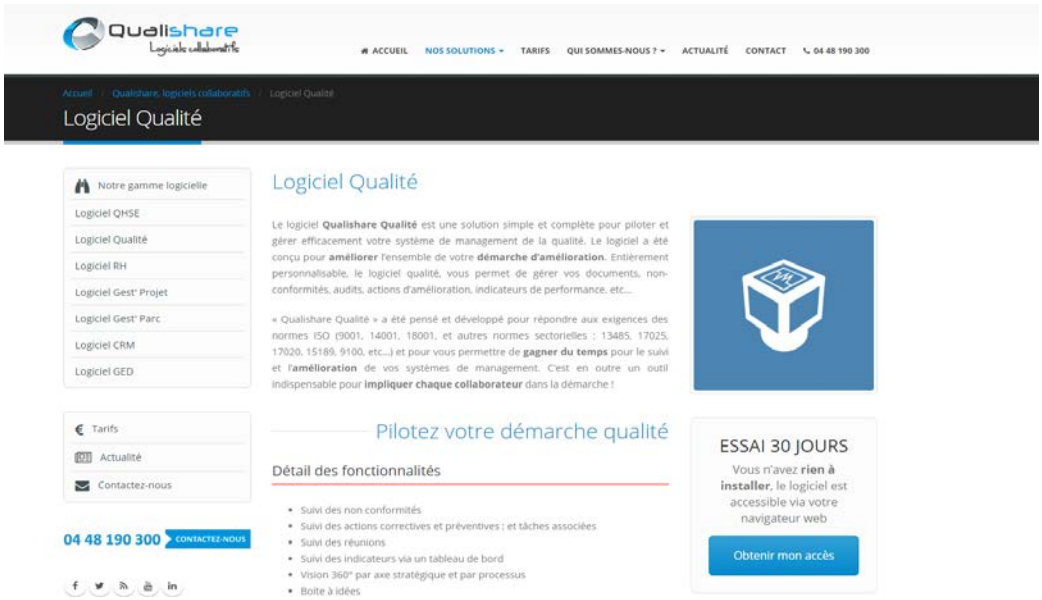

Fig.4. QualiShare Quality application interface [12].

- SMQ Agilium: SMQ AGILIUM is a Full Saas "Software as A Service" quality management software that enables the efficient management of audits, nonconformities, action plans, and business improvement requests. Thanks to visual indicators and dashboards, continuous improvement is managed in real time. The figure 5 shows an interface of the Agilium SMQ application.

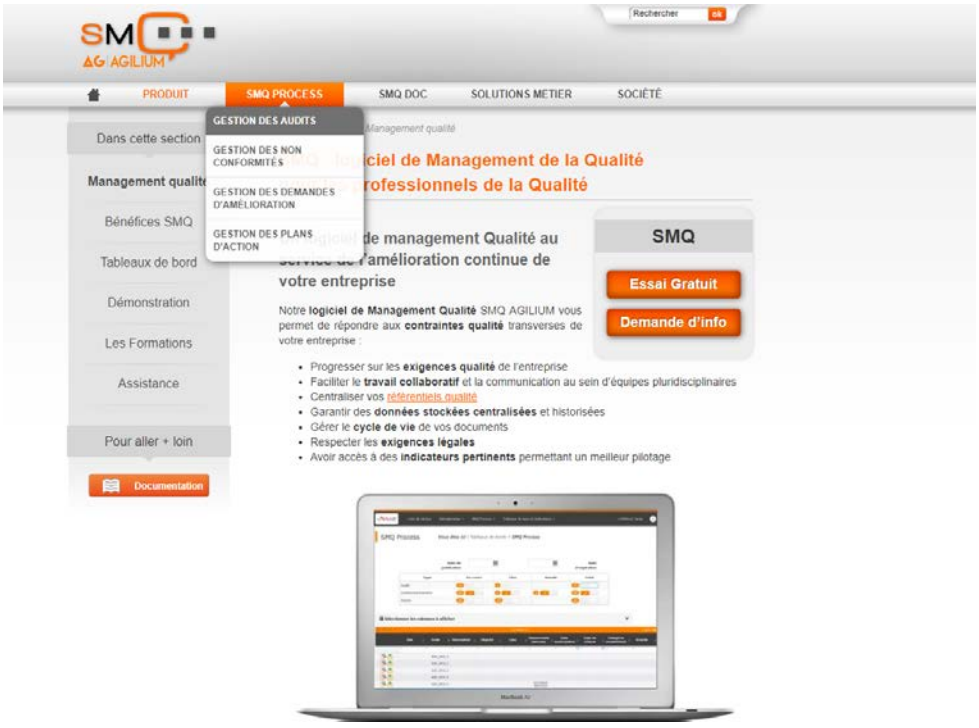

Fig.5. Agilium SMQ application interface [13].

2.5. Comparative study

What all three solutions have in common is the fact that they have ERP systems [17-20]. The table 1 shows a comparative study according to customer criteria between the solutions already presented in the previous section. We cite among others the work in [21] where the authors propose an Intelligent Machine Learning-based Business Project Management Process Approach that helps clients perform better and improve their project management process. In [22] where the authors propose an architectural model of the integrated system for the management of web-based e-commerce business processes. In [23] the authors propose a Context for Operations as a basis for the Construction of Ontologies to Employment Processes. The authors of [24] provides a business process management environment for collaborative research that can be supported by Web 2.0. The comparison table 1 is based on very important critical criteria on the client side that must always be taken into consideration.

In this section we have presented a view on the theoretical aspect of quality management and the importance of using a quality management system as well as the different solutions existing in the market.

Table 1. Comparative Table of The Presented Solutions

	Divalto Infinity Quality	**QualiShare Quality**	**SMQ Agilium**
Paid software.	Subscription 300 /month.	Price from 240 / month.	Personalized quote.
Complexity.	Light.	Heavy.	Heavy.
Type of company.	Software suitable for Small and Medium Enterprises (SMEs).	Is not specified.	Is not specified.
Fonctionality.	Heavy registration process.	No feedback for the RMQ.	Several fields to fill in and complexity in the interface.

The state-of-the-art analysis, presented in the work of [25], shows the use of budget achievement data and activity categories as comparators to determine activity status. Through this study, the authors also record the possibility of using IDX activity data, from 2018, to implement an integrated system. The results at the end of this study announce the achievement of the category and the budget of this activity strongly the activities which will then be affected or not. In another work [26], interesting from the literature, the authors study the introduction of BPR in the public sector. Thus, a framework is proposed to transform the change using a knowledge base. The results of the survey confirm that government agencies, the Saudi example, can gain the power to implement the BPR successfully, especially when the BPR movement started on a small scale and is implemented with the knowledge management. The authors in [27], in their proposed approach, create three organizational marketers in order to annotate the emotions and also to mark the feelings of each instance of the corpus. The results of the evaluation, returned by the simulation, show that this approach works better than humans. The study also found that the counselor's own emotions influence their perception of emotions by annotating the emotions and feelings in the datasets. And, in another interesting work [28], the authors propose an approach modeled by a multi-agent system (MAS) in the Fog-Cloud computing environment to simultaneously process

various QoS metrics, namely cost, makespan and latency. This approach has been validated by simulation with the workflowsim platform using different workflow types and system models based on Amazon EC2. All the results obtained demonstrate the effectiveness of this approach in generating optimal scheduling solutions in terms of QoS metrics and respecting the constraints.

3. Motivation

The many advantages of jBPM and its Integrity with Spring are the main motivations that have guided us to offer our new intelligent approach to integrating workflows into quality management systems. In the next two subsections, we will present the advantages of both as well as the easy integration of Spring into jBPM.

3.1. jBPM (Java Business Process Model)

jBPM stands for Java Business Process Management. It is a flexible business process management (BPM) suite written in the Java language. It helps to remedy the understanding failures between business analysts and developers. It is lightweight and completely open source in nature. This enables us to build, deploy, execute, and monitor business processes throughout their lifecycle. It is distributed under the Apache license and published under ASL by the JBoss company. The core of jBPM is a lightweight, extensible workflow engine written in pure Java. It allows us to run business processes using the latest BPMN 2.0 specification. It can be run in any Java environment and easily integrate into our application or as a service. The advantages of jBPM are as follows:

- jBPM is lightweight, fully open-source, and written in the Java language.
- jBPM allows you to model complex workflows using a graphic designer. The graphical designer helps non developers to design business processes and provides a better view of the state of a process at runtime.
- The jBPM workflow can also create tasks for human users. For example, manual testing or signing releases.

If we compare the difference between a traditional BPM and jBPM we can quote the following points:

- Traditional BPM process engines target only non-technicians, while jBPM process engines focus both technicians and non-technicians.
- jBPM is easily integrated into a Java project while traditional BPM systems require the installation of a separate server, which makes integration into the Java software development cycle difficult.
- jBPM supports several process languages while the display of BPM is not yet stabilized. There are currently many different interpretations of BPM leading to great market fragmentation.
- jBPM has very flexible transaction handling. If the application uses a JDBC connection in the Java environment, jBPM uses that JDBC connection to perform its task. If our application uses Hibernate, then jBPM can use the same Hibernate session factory. In case our application is running in a corporate environment, then jBPM can be linked to the surrounding JTA transaction while the BPM does not support these elements.

The next figure 6 illustrates the main features of jBPM.

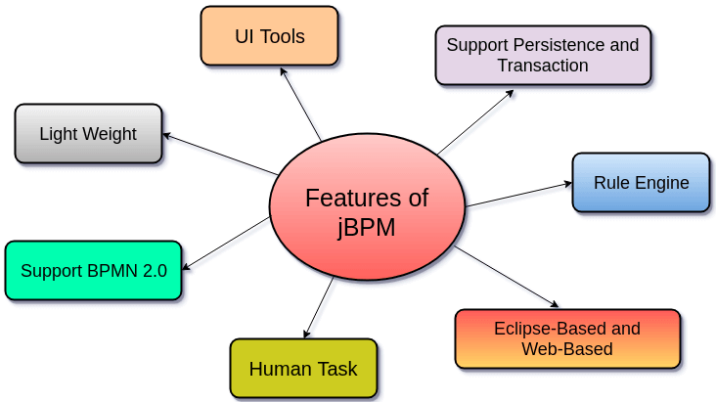

Fig.6. Main features of jBPM [14].

3.2. Integrity of jBPM with spring

Before getting into the integrity between jBPM and Spring, we will first define what is a Sping? Spring is one of the most popular application frameworks in the Java world. It allows great flexibility in the features and projects used in an application. It is, for example, possible to use the Spring container to manage beans in a basic way without using the

"AOP" aspects-oriented programming [16]. The purpose of Spring is to facilitate and make productive the development of applications and particularly business applications. Spring offers many basic features for application development. The most important in our solution is that it is able to import a lightweight container implementing the IoC design pattern for the management of objects and their dependencies by offering advanced functionalities concerning configuration and automatic injection. One of its strengths is to be non-intrusive in the application code while allowing the assembly of weakly coupled objects.

4. Proposed Workflow Integration in a QMS Approach

In this section, we present our proposed approach of integrating as a hybrid solution between the jBPM workflow engine and the Spring framework to guide the quality management processes. We also show the essential role of the Drools rule engine in the management of rules for a given process.

4.1. Integrated approach between jBPM and Drools

A rule engine is considered to be an interpreter of if/then statements. These if/then interpretations are called rules. jBPM provides a rules engine with a declarative language used to evaluate available information. The jBPM tools that allow the definition of rules are more or less advanced but generally still quite simple and few in number, among these rules we found:

- Boolean operators
- Process data fields
- Values entered by operators
- Properties of the documents possibly associated with the process, etc.

These rules are generally "included" in the definition of the process, and it is necessary to version the latter to modify them, which requires administrative operations on the workflow engine. To remedy this problem, jBPM integrates in its kernel the Drools rule engine which allows to define very elaborate routing rules. It is complementary to the workflow engine, and it allows users to be exposed to simple interfaces for generating rules. jBPM can then connect to the rules engine to "know" which route to take during a process.

The strong point of jBPM is integrity with Drools, the 2 open sources are now closely associated and fully support the 2 sister standards BPMN and DMN (Decision Model and Notation) which present the two respective modeling languages. The next figure 7 illustrates the interoperability relationship between the jBPM rule engine with the BPMN specification and the Drools rule engine with the DMN notation.

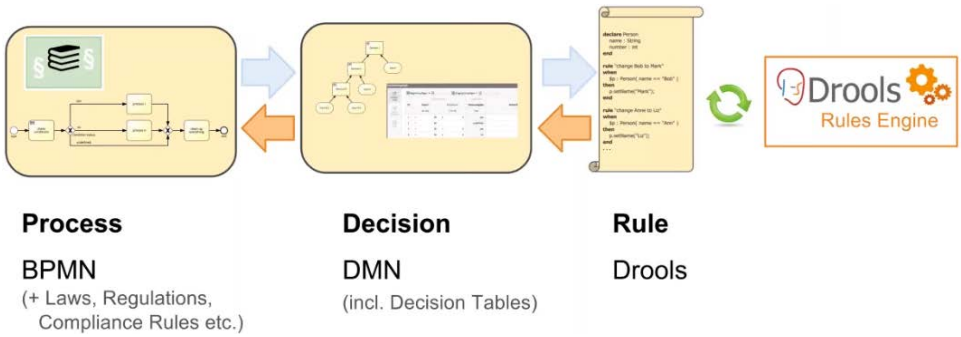

Fig.7. Interoperability between jBPM and Drools [15].

4.2. QSE business rule modeling

In this part we will explain the business rules integrated in our solution to choose the module requested by the user. Figure 8 shows a complete example on the DMN business rule modeling standard and inter-operability with the BPMN business process.

The business process implemented in our application aims to manage user requests and divert them according to their choice and for which module will interact. This will be described in the following components:

- The “Business Rules” decision-type task calls a “Routing Decision Service” decision service by transmitting the requested module as input. The service produces four decisions.
- A gateway uses the value of "Business Rules" to direct itself to "Doc Manage", "Audit Manage "," Execute Review "or" C&P Actions "tasks.
- The “Doc Manage” task gathers document and template data.

- The "Audit Manage" task gathers data from audit reports, the annual audit plan and audit reports.
- The “Execute Review” task gathers annual management review data.
- The "C&P Actions" task collects data from corrective and preventive actions throughout the year on a cyclical basis.

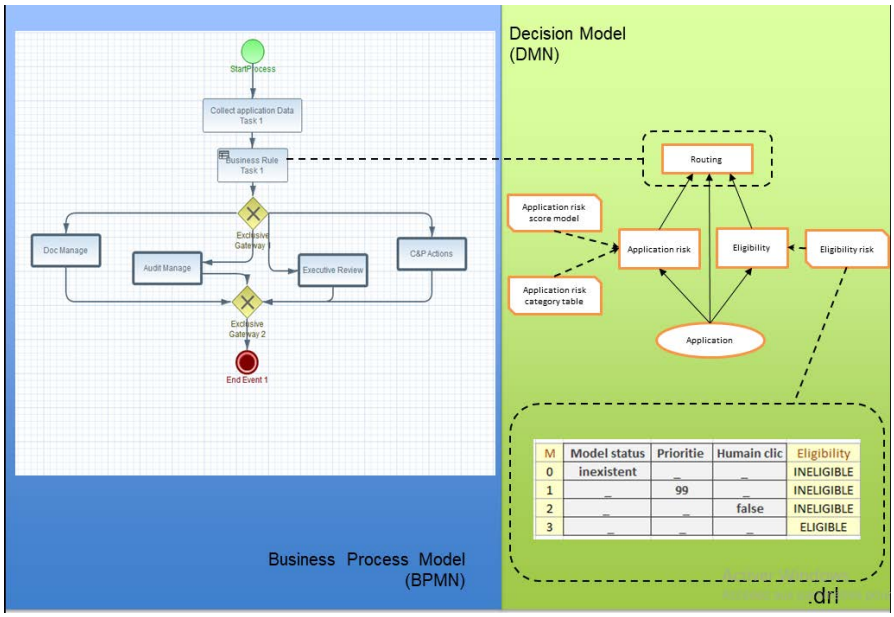

Fig.8. Proposed modeling architecture of business rules.

Regarding the decision table, it is a tabular representation of a set of expressions, organized in business rules indicating which input data will apply to the system. It contains all of the expressions necessary to determine the output data. In addition, it is a complete table that contains all possible combinations of expressions. The 3 cells in red in the decision table in figure 8 represent the following rules:

- If the value of "Model status" satisfies the "NONEXISTING" condition, then whatever the other parameter values in input, the final rule already satisfied, and the result of the decision table is "INELIGIBLE" in output.
- If the value of "Priority" satisfies the condition "less than 1 and greater than or equal to 5", therefore whatever the other parameter values in input, the final rule already satisfied, and the result of the decision table is " INELIGIBLE "in output.
- If the value of "Human click" satisfies the condition "false", then whatever the other parameter values in input, the final rule already satisfied, and the result of the decision table is "INELIGIBLE" in output.
- If the value of "Model status" satisfies the condition "EXISTENT" and the value of "Priority" satisfies the condition "greater than or equal to 1 and less than 5" and the value of "Human click" satisfies the condition "true", then the rule is satisfied, and the result of the decision table is the "ELIGIBLE" output data.

Now moving on to the decision-making diagram. The decision service required for the business process is modeled as shown in the figure 9:

The "Model constraint" block present in the DMN, the Business Knowledge or the "BKM", models the knowledge as a function of parameters defined in "input Data" in the form of a decision table (already declared above). This table is used to write the logic of a decision in the form of a reusable function, "it is the logic of knowledge": "Business Knowledge Models should only be applied when an actual reuse scenario is imminent." « James Taylor and Jan Purchase ».

Regarding the "Supporting document for model" block, it is a "Knowledge Source" type block, which allows access for our "BKM". It should be taken into account when making a decision or operating a business model. The link between this block and the "Model constraint" block is called an authority requirement.

The “Routing Decision Service” service, called by the “Business Rules” task, presents as output the “Routing” decision which is the result of the “Model constraint” decision logic. The link between this block and the "Model constraint" block is called a knowledge requirement.

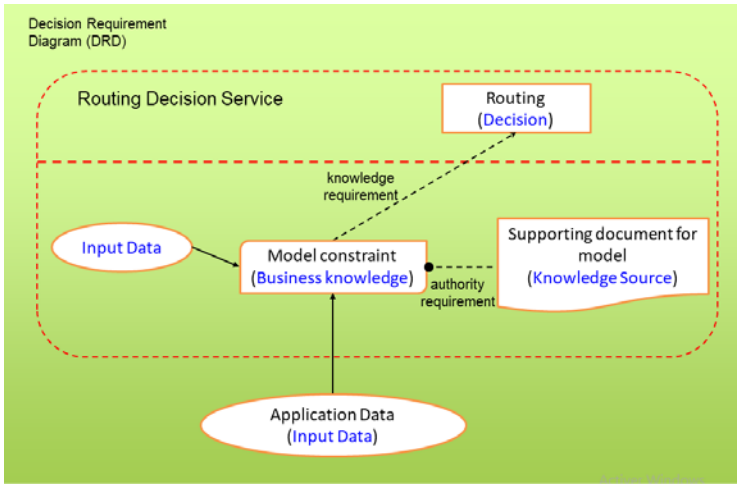

Fig.9. Diagram of Decision Making.

4.3. Logical architecture of our solution

jBPM is one of the workflow engines that does not support native spring like cammunda and activiti in its kernel due to the dependency conflict between Spring and Drools. In our application we need a good rule manager because the quality management system has full processes to manage it and Drools performs best in this case. On the other hand, Spring is one of the most robust and flexible frameworks in terms of its functionality. So the solution we adopted in this case is to integrate Spring with jBPM without excluding Drools.

Spring's integration with jBPM is done through REST APIs (It is an application programming interface that uses HTTP requests to get, place, publish and delete data between heterogeneous applications). The following figure 10 illustrates the overall architecture integrated into our solution.

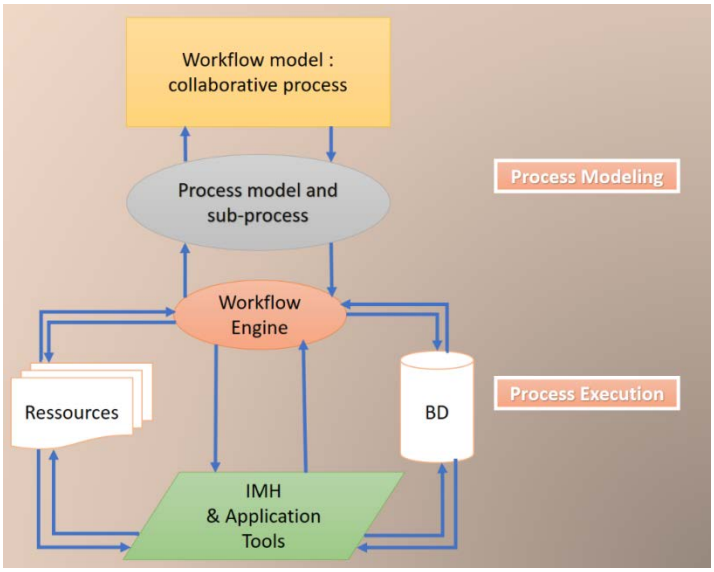

Fig.10. Global architecture of the proposed approach.

As we stated previously, the communication between jBPM and Spring is done through Web services. This is because jBPM offers the ability to create and use custom and domain-specific nodes in our business processes. This simplifies development when we are going to create processes containing tasks specific to our quality management system.

And, since REST web services represent an evolution in the IS world, we have decided to customize "Customer TASK" type nodes which will consume the web services exposed by our Spring application. These nodes will dynamically generate the client corresponding to the URL and invoke the operation as a parameter. The steps for creating these tasks are as follows:

- Creation of "custom work items": Adding a node with a custom work item to a process definition by configuring a definition.wid file based on the MVEL language. The extension wid is the abbreviation of "work item definition".

SMQDefinitions.wid | drools.rulebase.conf

```
import org.drools.process.core.datatype.impl.type.StringDataType;
import org.drools.process.core.datatype.impl.type.ObjectDataType;
import java.lang.String;
[
  [
   "name" : "WSRestForSMQ",
   "description" : "Générer dynamiquement le client correspondant à l'URL et invoquer l'opération en  paramètre.
    Les attributs EndPoint Operation et Interface vont contenir les noms des variables qui contiennent les valeurs
     à passer.Le paramètre interface va contenir la version du webservice ",
   "parameters" : [
        "CommandClass" : new StringDataType(),
        "EndPoint" : new StringDataType(),
        "Operation" : new StringDataType(),
        "inputParamName":new StringDataType(),
        "resultVariable": new StringDataType(),
        "exceptionMessage": new StringDataType() ,
   ],
   "displayName" : "WSRestForSMQ",
   "icon" : "../icons/wsGenerique.jpg"
  ],
]
```

Fig.11. Configuration of the new REST web service node.

- Synchronize this node with the Drools rule engine to find out when running one of the appropriate schemas for this node.

SMQDefinitions.wid | drools.rulebase.conf

```
drools.workDefinitions=SMQDefinitions.wid
```

Fig.12. Synchronizing the node with the Drools rule engine.

- Creation of "custom work item handlers" so that the jBPM engine in our process during execution, executes the code of this handler: it is java code.

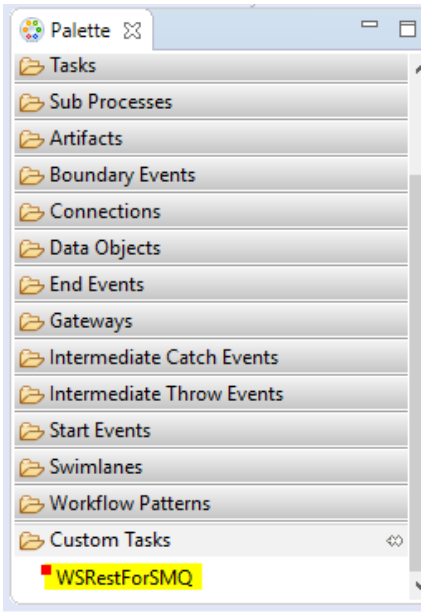

Fig.13. Adding the new node in the node's palette.

4.4. Module choice algorithm

To fully explain the different steps of our approach, we propose the algorithm described by Algorithm 1.

Algorithm 1 Choise Module

```
Require : ProcessInstance p ;
          Module m  ;
          Node i ;
          HashMap<String, String> mapParam ;
Begin
    p=lancer(m);
    for(i = 0 : p.sizeNode() ){
        if( ifHumainTask(i) ){
            while( !waitClick() ) {}
        }
        if( isCustomerTask(i) ){
            invokeRest(mapParam);
        }
        trigger(i);
    }
Return p.status ;
End ;
```

```
Function lancer (Module m ) : ProcessInstance

Require : StartProcessRequest startProcessRequest;
          StartProcessResponseType startProcessResponseType

Begin
    startProcessRequest.setLogin(m.login);
    startProcessRequest.setPassword(m.password);
    startProcessRequest.setSystemOwner(m.systemOwner);
    startProcessResponseType =
    getBPMToolsWSSOAP().startProcess(startProcessRequest);
    Return startProcessResponseType.getProcessInstanceId();
End
```

```
Function invokeRest (HashMap<String, String> mapParam)

Require : Header header ;
          URL url ;
          String sysConfig;
           ServiceRest service;
Begin
    header = mapParam.get (" header ");
    url = mapParam.get ("url");
    sysConfig = mapParam.get (" sysConfig");
    service = SingletonWrappers.getWsInstance( new URL(url) , header , mapParam );
    service.setCodeParam(mapParam.get ("codeParam"));
    service.invokeMethode(mapParam.get ("operation") );
End
```

```
Function invokeRest (HashMap<String, String> mapParam)

Require : Header header ;
          URL url ;
          String sysConfig;
           ServiceRest service;
Begin
    header = mapParam.get (" header ");
    url = mapParam.get ("url");
    sysConfig = mapParam.get (" sysConfig");
    service = SingletonWrappers.getWsInstance( new URL(url) , header , mapParam );
    service.setCodeParam(mapParam.get ("codeParam"));
    service.invokeMethode(mapParam.get ("operation") );
End
```

We detail each of the functions on which our application is based.

- launch (Module m): allows you to launch a module chosen by the user from the Dashboard of our solution and it returns a process instance specific to this action. To develop this idea well, these figures show the detail of this function.
- ProcessInstance.sizeNode (): returns the number of nodes in a process.
- ifHumainTask (Node n): allows you to check if the type of this node is a Human Task or not.
- isCustomerTask (Node n): allows you to check if the type of this node is a customize node or not, and since we have identified a single personalized node which is used to manage the integration with the Spring application.
- ProcessInstance.status: the status attribute of the Process Instance is used to identify the status of the process, either blocked, completed, canceled or active.
- invokeRest (HashMap map): this function takes a data map as a parameter; it is used to consume REST-type APIs. Function 2 illustrates this function.
- Trigger (Node n): this function consists of executing the processes behind the nodes of the process. Function 3 summarizes the details of this function.

In this section, we have presented the integrity approach of a workflow engine with quality management processes, and we have analyzed the integrity between jBPM, drools and the Spring framework to deal with quality management modules.

5. Experimental study and results analysis

In this section, we will detail the realization part of our solution and we will discuss the results found. After having presented, in the previous sections, the notion of quality management system and the notion of a workflow engine as well as the interaction between the rule engine and the spring framework. This section will be devoted to the presentation of the development tools used with the results obtained during the integration tests.

5.1. Environment and experiments

In this section, we present the technical environment as well as the justification for various technological choices. To choose the appropriate programming language to develop our solution, we carried out research on the most used programming languages given their adequacy with the theme addressed in the framework of our research master's project. According to the survey done by the Stackoverflow site (it's an ultra-popular site where programmers ask questions and anyone can help with code problems), we found that the most popular languages are JavaScript, Sql and java. The following figure 14 illustrates the result of the poll.

From the results of this survey, we noticed that the most widely used programming languages are: JavaScript, Java, and SQL. So, we used;

- Javascript language for the integration of the Front-End part. It allows the design of dynamic interactive websites and applications running on all web browsers, be it Google Chrome, Mozilla Firefox or Safari. In addition, the very active community and many tools and modules are available to use this language facilitate its use and allow constant innovation.
- The java language for the integration of the Back End part. Java's virtual machine, the JVM, allows the same Java program to be used on a wide range of platforms, so our workflow engine is based on the java language.
- SQL for integration with the database. In fact, the majority of database projects are based on SQL (90% of the database market is monopolized by relational databases).

Framework Angular

Angular is an Open-Source JavaScript framework developed by Google. It uses the MVM (Model View Model) architecture. It will allow you to structure your code and separate the view (the interface) from the models (operation).

It is considered a "client-side" language that allows the user interface of each page (display, interactions...) To be managed dynamically and complements server-side languages. We used the latest version in the market: Angular 7. This screenshot shows the interface of our solution created by the Angular framework.

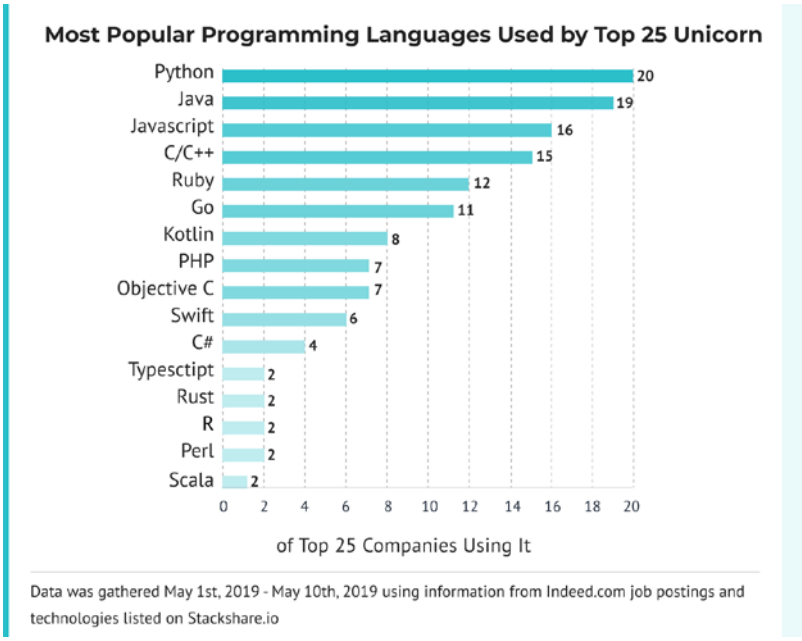

Fig.14. Results obtained by the survey of the Stackoverflow site [16].

Fig.15. Angular interface of our solution.

5.2. Spring DATA JPA

Spring Data JPA, part of the large Spring Data family, makes it easy to implement JPA-based repositories. This module covers enhanced support for JPA-based data access layers. This makes it easier to build applications using Spring, which use data access technologies.

Implementing a data access layer of an application is a tedious task. Too much master code has to be written to perform simple queries and to perform paging and auditing. Spring Data JPA aims to significantly improve the implementation of data access layers by reducing the effort to the level actually required. We're writing our repository interfaces, including custom search methods and Spring will automatically take care of the implementation.

5.3. Find Bugs

Find Bugs is an open-source plugin that helps identify flaws and bugs in the source code of our application. It is very scalable and allows you to create a plan to identify all the faults and classify them by priority according to their influence on the operation of the application.

5.4. JUnit

JUnit is a unit testing framework for the Java programming language. We used the assertions to test the different methods created in the back-end part. To simulate the test well, we used Mockito which allows us to create and configure virtual objects to simplify test development for classes with external dependencies.

5.5. Postman

It is a multi-functional tool for Web APIs that allows you to build, execute HTTP requests and store logs so that they can be re-executed. It also manages environments allowing variables to be contextualized and to execute queries or

series of queries in different configurations (typically: dev, recipe, prod). This tool allows us to test the different APIs created at the level of our back end. Figure ref postman shows the interface of this tool.

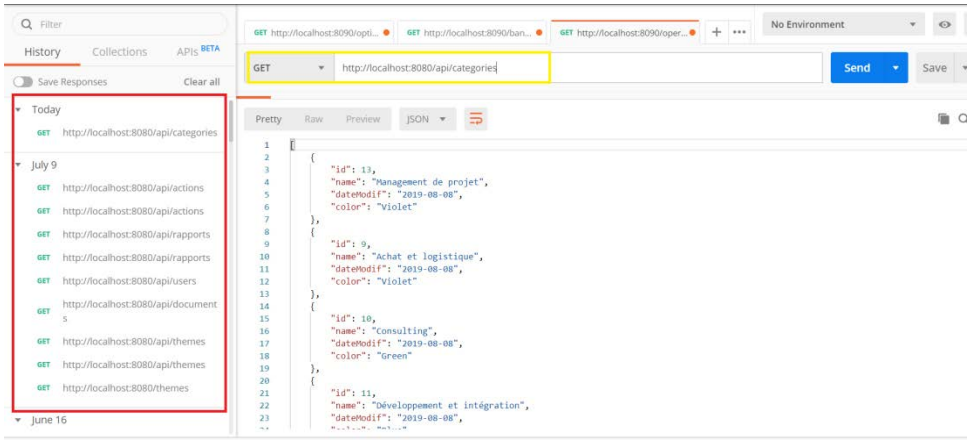

Fig.16. Testing REST APIs on the postman tool.

5.6. Experimenting and Results analysis

In this part, we will show the different results obtained concerning each integrated module for quality management. We cannot speak of a quality management system without speaking either of the management of the various documents, or of the management processes or the models of modeling and organization. At the level of the application carried out, our dashboard takes care of the structured presentation of these documents, organized by category, because according to the ISO 9001 standard, documents are of several types: logistics purchasing documents, documents project management. The interface represented by the figure ref doc shows the organization of these documents by type of category.

Fig.17. Interface for viewing the list of categories.

The process on the jBPM side, behind this processing, is represented by the figure 18. When the RMQ chooses a document, the Spring application notifies jBPM with a new action, the latter retrieves the current object and starts a new process identified by its unique key "ProcessInstanceId". A web service is invoked to load a document and invoke one of the CRUD methods (read, update, delete) or to add a document. Figure 18 shows the document management process at the jBPM application level. This process is made up of several nodes and Gateways to divert the process according to the management rules.

The next figure 19 shows the content of the "select document" sub-process.

The figure 20 shows statistics concerning the documents managed by our application. Indeed, this table summarizes the classification of documents by category and gives a view of the number of documents in each category.

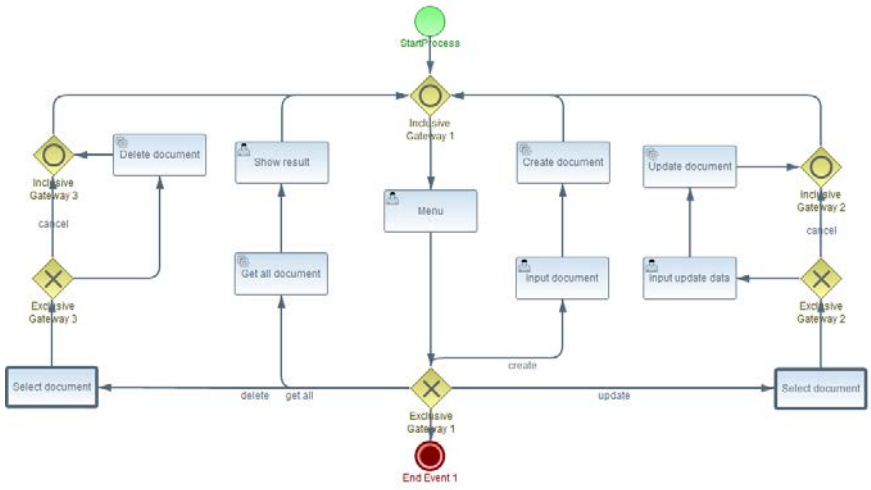

Fig.18. Document management process.

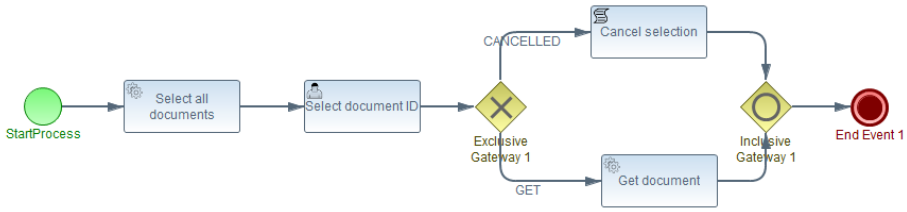

Fig.19. Document selection process.

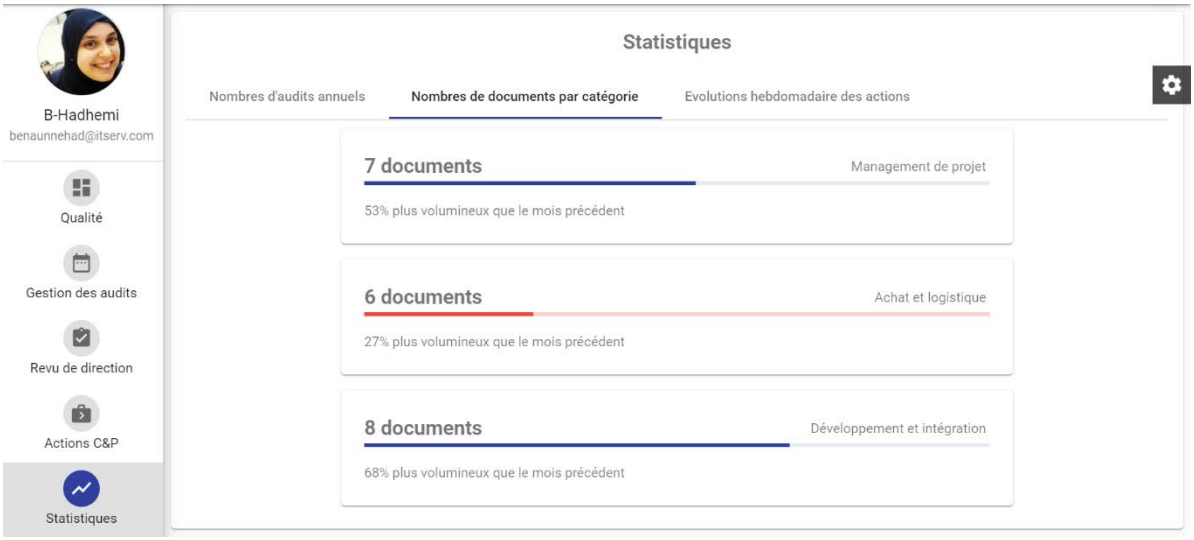

Fig.20. Document statistics managed by category.

5.7. Audit and results management module

In this part, we focus on the part relating to audits. For IT SERV, this procedure describes the management of internal quality audits. Its objective is to define the organization, planning, performance and monitoring of internal audits covering all the activities of a company's quality system. The aim is to ensure:

- Effective monitoring of the quality system (its conformity, efficiency and improvement).
- Correction of deviations and / or improvement of the system, identified during the audit.
- Relevant information to management of the results of this monitoring and the corrective actions and/or improvements initiated.

The audits are carried out under the responsibility of the quality manager. The documentary support for the audit procedure is made up of 3 processes:

- Annual planning of audits.
- Audit plan.
- Audit report.

These 3 processes are illustrated in the interface presented by figure 21 for the audit procedure.

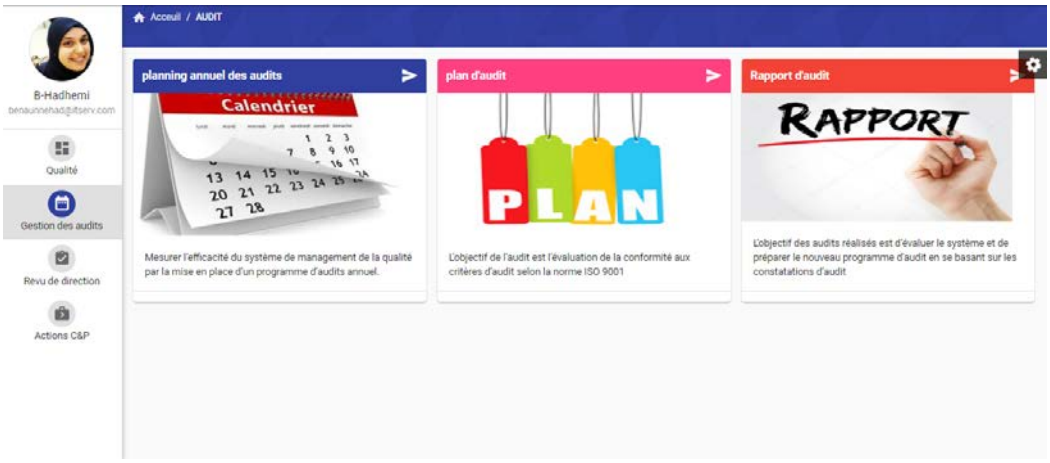

Fig.21. Documentary support interface for the audit procedure.

We also presented real-time statistics on completed audits and pending audits, as shown in figure 22.

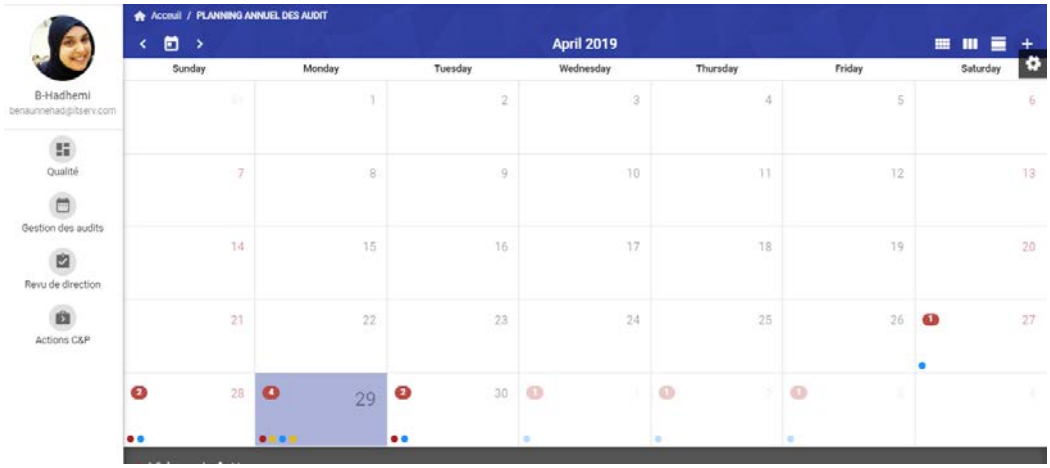

Fig.22. Results of annual audit statistics.

Fig.23. Weekly evolution of the number of corrective and preventive actions.

5.8. Evolution of preventive and corrective actions

Our company must continually improve the relevance, adequacy, and effectiveness of the quality management system through improvement and monitoring actions. Indeed, the RMQ is responsible for recording on the action tracking file, in order to carry out reminders, when necessary. Depending on the value of the type of finding, our rule engine

determines the priority of this action. Carrying out statistics makes it possible to estimate the performance indicator, for this the graph corresponding to the figure 23 shows the weekly frequency of actions.

5.9. JBPM performance and task evolution

The learning curve of jBPM is very high due to the complexity. jBPM does not offer a service task as suggested by the BPMN only standard, but we can define our own Java service tasks as "WSRestForSMQ" and manually register them in the engine, this allows us to do some fairly low-level programming. Figure 24 shows the evolution of the number of service tasks in the last four versions of our workflow engine compared to other workflow engines.

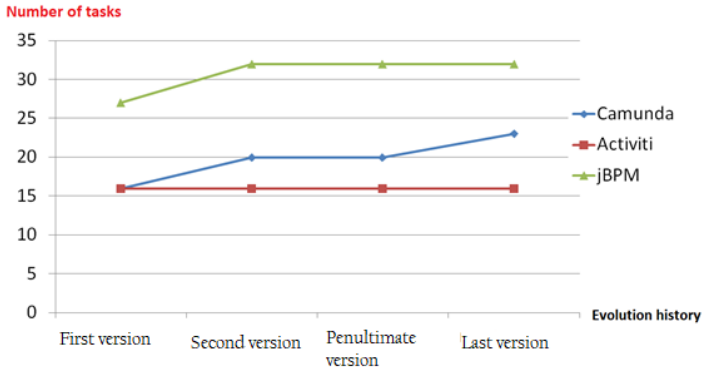

Fig.24. Evolution of the number of tasks in the last four versions.

Currently, jbpm occupies 33 declared Task Services not to mention the scalability of this engine allowing to customize tasks according to the customer's need. As for our solution, the statistics show the performance of our new task in terms of calling the Rest web services from the Spring application. Without canceling or blocking the process, she completes her treatment perfectly. Figure 25 shows the number of tasks already in use per process instance.

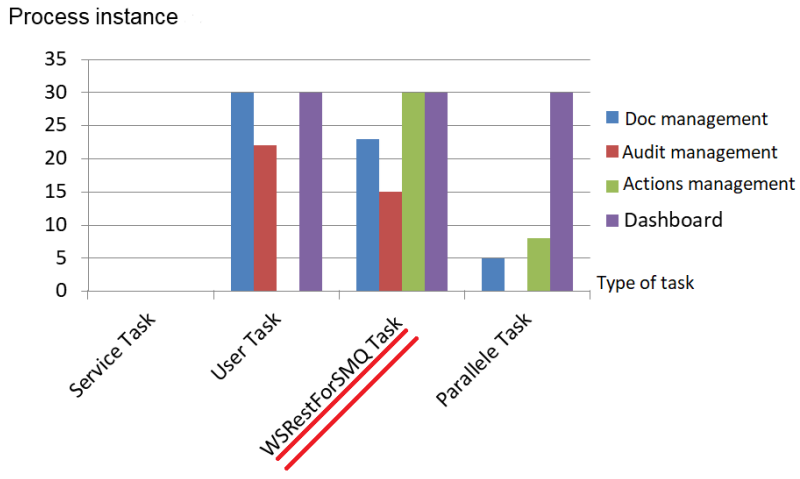

Fig.25. Number of tasks used per process instance.

6. Discussion and Limits

The objective of our work, in this article, is to propose a new approach to workflow integration in a quality management system. In this perspective, we presented the general framework of the project, namely, the context of the project, the presentation and the overall analysis of the different aspects of a quality management system as well as a review of the state of the literature of the main existing solutions. Through this study, we were able to identify the limits and advantages of each of the existing solutions and noted the importance of a workflow engine. This study also allowed us, among other things, to choose the modeling language of the BPML process. We succeeded in presenting a new solution which consists of an approach, on the one hand of integration between the jBPM workflow engine and the different processes and, on the other hand, of communication with the Spring framework and the Drools rule. The choice of the working environment and the different technologies used when developing a solution is a task of paramount importance for the success of such a project.

The results we obtained show the importance of the proposed solution, its feasibility and confirm its performance through the statistics presented in the experimental part.

Certainly, our work remains open to several perspectives, given that no work is a finished work. Indeed, our solution has certain limitations such as over-exploited resources. Thus, we intend to develop and integrate a collaborative decision support module in order to optimize system resources. We also intend to enrich our approach with the mechanism of "Data Science" in order to bring more efficiency to our solution.

7. Conclusion and Future Works

7.1. Summary

The objective of our work is to propose an intelligent approach of integrating workflows into a quality management system. We have shown, through the different simulation results, the feasibility of our solution as well as its efficiency compared to the different approaches existing in the state of the art. We started by presenting the general framework of our approach, namely, the general context, the presentation and the global analysis of the different aspects of a quality management system as well as the main solutions existing in the literature. Through the summary of the main approaches that we studied we were able to choose the BPML process modeling language. The next phase was devoted on the one hand to the presentation of the integration process between the jBPM workflow engine and the different processes and on the other hand to communication with the Spring framework and the Drools rules engine. And we presented the work environment as well as the different technologies used during the simulation of our solution. Finally, the results obtained at the end of the simulation of our approach were analyzed and performance statistics were also presented.

7.2. Prospects

Of course, our approach remains open to several perspectives. Our future work will therefore be structured around three directions. The first direction is to conduct a more in-depth comparative study between the main approaches studied in the literature in order to give more amplification to academics and practitioners on how to manage Workflow integration in a Quality Management System. The second direction consists in enriching our approach by the mechanism of "Data Science" to be able to bring more efficiency to our solution. We also plan to offer a collaborative decision support module to optimize system resources as a third direction.

References

[1] Organisme internationale de certification. "NF EN ISO 9001 Octobre 2015", Octobre 2015.
[2] Moulay Abdelah Amadour. "la qualité comment s'y prendre au Maroc", Université Moulay Ismail de Meknès, 2000.
[3] Arnaud Eve et Pierre-Antoine Sprimont. "Perceptions et attitudes liées à la norme ISO 9001 : une analyse auprès de salariés opérationnels", Association Francophone de Comptabilité, décembre 2015.
[4] JORGE CARDOSO. "Workflow Management Systems vs. ERP Systems", Département de mathématiques et d'ingénierie, Université de Madère, 9050-078 Funchal, Portugal, 2006.
[5] David D. Coleman et Morgan Kaufman. "Requirements for workflow products", University of Twente, The Netherlands, Aout 1992.
[6] JORGE CARDOSO. "Workflow Management Systems vs. ERP Systems", Département de mathématiques et d'ingénierie, Université de Madère, 9050-078 Funchal, Portugal, 2006.
[7] BENDOUKHA Hayat. "Une Approche Workflow pour la Définition et la Composition des Services dans une Grille de Calcul", Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf- Algérie, Juin 2010.
[8] Robert M Shapiro. "A Technical Comparison of XPDL, BPML and BPEL4WS", 12/8/2002.
[9] Michal BALI. "Drools JBoss Rules 5.0", 2009.
[10] Julien Da Costa. "BPMN 2.0 Pour la modélisation et l'implémentation de disositifs pédagogiques orientés processus". Université de genève, 2014.
[11] Divalto Divalto logiciel de gestion de qualité : Disponible sur le site : http://www.mediasoft.com.tn. [Accès le 01-Mars-2018].
[12] QualiShare QualiShare logiciel de gestion de qualité : Disponible sur le site : https://www.qualishare.fr. [Accès le 01-Mars-2018].
[13] Agilium SMQ Agilium logiciel de gestion de qualité : Disponible sur le site : http://smq.agilium.com. [Accès le 01-Mars-2018].
[14] jbpm Documentation. Documentation officiel du moteur de workflow jBPM : Disponible sur: https://docs.jboss.org/jbpm/v6.4/userguide. [Accès le 30-Aout-2018].
[15] spring Documentation officiel de Spring tool suit : Disponible sur le site: https://spring.io/tools/sts. [Accès le 27-Aout-2018].
[16] Plateforms FlyAPS SOFTWARE COMPANY : Disponible sur le site: https://flyaps.com/blog/top-10-coding-languages-used-by-global-companies/. [Accès le 30-Juilet-2019].
[17] stack Plateform open source pour les programmeurs : Disponible sur le site: https://stackoverflow.com/ . [Accès le 30-Juilet-2019].
[18] Yasunao Takano, Yusuke Iijima, Kou Kobayashi, Hiroshi Sakuta, Hiroki Sakaji, Masaki Kohana, and35Akio Kobayashi. Improving document similarity calculation using cosine-similarity graphs. International Conference on Advanced Information Networking and Applications, 926(Springer):512–522, 2019.
[19] Ton Tran, Wen Cheng Liu, Bernard K. Chen, Andrey Molotnikov, Xinhua Wu, and Jerome Pun. Ctbased38agile 3d printing system for pre-operative planning of orthopedic surgeries. International Journal of Agile Systems and Management, 13(1):1–

27, 2020.
[20] Xuegong Zhang. Using class-center vectors to build support vector machines. Neural Net-works for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop, Cat. (No.98TH8468):3–11, 2019.
[21] Oleg Pursky, Dmytro Mazoha," Architecture Model of Integrated Web-based E-trading Business Process Management System", International Journal of Information Engineering and Electronic Business, Vol.10, No.2, pp. 1-8, 2018.
[22] Salah Alkhafaji, B. Sriram, "Conceptualization and Integration of Information Systems in Educational Business Activities", International Journal of Information Engineering and Electronic Business, vol.5, no.2, pp.28-33, 2013.
[23] Iryna Zavuschak, Yevhen Burov, "The Context of Operations as the basis for the Construction of Ontologies of Employment Processes", International Journal of Modern Education and Computer Science, Vol.9, No.11, pp. 13-24, 2017.
[24] Asli Sencer, Meltem Ozturan, Hande Kimiloglu,"A Web 2.0 Supported Business Process Management Environment for Collaborative Research", International Journal of Information Engineering and Electronic Business, vol.7, no.1, pp.8-17, 2015.
[25] Ali Fajri, Ardiles Sinaga, "Implementation of Business Intelligence to Determine Evaluation of Activities (Case Study Indonesia Stock Exchange)", International Journal of Information Engineering and Electronic Business, Vol.12, No.6, pp. 51-67, 2020.
[26] Arwa S. Bokhari, Rizwan J. Qureshi, “Business Process Re-Engineering in Public Administration of Kingdom of Saudi Arabia", International Journal of Information Engineering and Electronic Business, Vol.8, No.4, pp.10-17, 2016.
[27] Stephen Opoku Oppong, Dominic Asamoah, Emmanuel Ofori Oppong, Derrick Lamptey, "Business Decision Support System based on Sentiment Analysis", International Journal of Information Engineering and Electronic Business, Vol.11, No.1, pp. 36-49, 2019.
[28] Marwa Mokni, Sonia Yassa, Jallel Eddine Hajlaoui, Rachid Chelouah, Mohamed Nazih Omri, Cooperative agents-based approach for workflow scheduling on fogcloud computing, Journal of Ambient Intelligence and Humanized Computing (2021) 1–20.

Authors’ Profiles

Mohamed Nazih OMRI received his Ph.D. in Computer Science from University of Jussieu, Paris, France, in 1994. He is a professor in computer science at the University of Sousse, Tunisia. From January 2011, he is a member of MARS (Modeling of Automated Reasoning Systems) Research Laboratory. His group conducts research on Information Retrieval, Data Base, Knowledge Base, and Web Services. He supervised more than 20 Ph.D. and Msc students in different fields of computer science. He is a reviewer of many international journals such as Information Fusion journal, Psihologija Journal, and many International Conferences such as AMIA, ICNC-FSKD, AMAI, SOMeT, etc.

Hadhemi BEN AONNE did her higher studies at the National Engineering School of Sousse in Tunisia where she obtained her engineering degree in Computer Science option Applied Computer Science. Currently she is a software engineer at Sofrecom Tunisia and Phd student in Higher School of Communications of Tunis.

How to cite this paper: Mohamed Nazih OMRI, Hadhemi BEN AONNE, "Towards an Intelligent Approach to Workflow Integration in a Quality Management System", International Journal of Intelligent Systems and Applications(IJISA), Vol.14, No.3, pp.54-73, 2022. DOI: 10.5815/ijisa.2022.03.05