International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 10 (2018) pp. 238-251
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Received : 8 February, 2018; Accepted : 27 August, 2018; Publish : 5 September, 2018

On the Automation of Managing and Retrieving Versions of Subsea Pipeline Installation Process

Fatma Ellouze[1], Mohamed Amine Chaâbane[1], Eric Andonoff[2] and Rafik Bouaziz[1]

[1] University of Sfax, MIRACL,
FSEGS, Airport Road, BP 1088, 3018 Sfax, Tunisia
fatma.ellouze@fsegs.rnu.tn, mohamedamine.chaabane@isaas.usf.tn, rafik.bouaziz@usf.tn

[2] IRIT, University of Toulouse 1- Capitole,
Rue du Doyen Gabriel Marty, 31042 Toulouse, France
Eric.Andonoff@ut-capitole.fr

***Abstract*: Subsea Pipeline Installation (SPI) business process depends on several contextual parameters, such as the weather conditions, the sea depth and the pipeline length. Due to this context dependency, several versions of SPI process have to be defined each corresponding to the right process according to contextual parameter values. Consequently, companies involved in SPI process face some challenges in such a multi-version environment. In fact, they must manage versions and retrieve the appropriate SPI process version according to a given context. These challenges are further complicated due to the lack of guidelines that assist them in this delicate task. In this paper, we recommend a new software tool called "Version Management and Retrieval" (VMR) to automate the management and the retrieval of process versions and illustrate his automation within the SPI process. On the one hand, VMR assists companies in the definition, the updating and the deletion of versions of SPI and on the other hand, it enhances each SPI process version with semantic aspects featuring its use context. This semantic aspect is helpful to retrieve the appropriate SPI process version from a given situation. The experimental evaluation of VMR shows promising results.**

Keywords: Subsea pipeline, Business process versions, Context, Semantic aspect.

I. Introduction

The installation of new subsea pipelines has received much attention all over the world. These pipelines are used primarily to carry oil or gas, but transportation of water is also important. Many research studies have been conducted to address issues, such as construction, maintenance, integrity, and repair of pipelines [1] [2]. In this paper, we argue that issues relevant from the maritime area, and more particularly the installation of new subsea pipelines, should benefit from Business Process Management (BPM) area. Indeed, BPM has gained adoption in a huge number of companies as it ensures consistent outcomes and takes advantage of improvement opportunities for process management and execution [3] [4] [5]. As a consequence, the central question addressed in this paper is "how can BPM contribute to a better execution of the Subsea Pipeline Installation (SPI) process?"

While SPI process runs smoothly for most of the year, many contextual parameters appear and can disturb its functioning. Indeed, the changes related to the weather conditions, sea depth, diving methods, IMCA (International Marine Contractors Association) norms, and so on, drive the SPI process owner to define new versions to adapt its process to these changes. After analyzing the SPI process, we have found more than 60 versions for this process, each corresponding to the right process according to contextual parameter values. These contextual parameter values define the use context of SPI process versions and these versions are alternatively executed according to the current situation. Due to the high number of version, which can increase again, the process owner is facing a real problem in managing them. A related problem is the selection, among these different versions of process, of the most appropriate one from a current situation. This issue is raised both for the process owner and his stakeholders who desire to coordinate their processes with an appropriate version of SPI process.

Therefore, the purpose of this paper is to introduce a software tool to improve SPI process version management, and retrieve adequate versions of SPI process.

The paper contribution is threefold. Firstly, we introduce the different SPI process versions as well as a solution to semantically describe each version of SPI process using contextual parameters. Secondly, we demonstrate the feasibility of our solution through the VMR (Version Management and Retrieval) tool, (i) which enables to define, update and delete SPI process versions, and (ii) also ensures semantic exploitation of versions of SPI process to retrieve the appropriate one according to a given context. Finally, we evaluate VMR tool by using F-measure, precision and recall metrics.

Accordingly, the remainder of this paper is organized as follows. Section 2 reviews the related work on subsea pipeline and version management. Section 3 shows how to model SPI process using a version-based approach. Section 4 is dedicated to the presentation of our solution that consists in managing and retrieving versions of processes. It also presents the feasibility of our solution through the VMR software tool. Section 5 provides measures that help evaluate the VMR tool using precision and recall metrics. Finally, Section 6 provides the conclusion and gives some directions for future research.

II. Related Work

Marine science is a broad discipline. Hence, numerous papers can be found in broader journals, such as the journal of Marine Policy, the journal of Marine Structures, the journal of Ships and Offshore Structure, the journal of Maritime Engineering, the journal of Engineering for the Marine Environment, the journal of Pipeline Engineering as well as a number of international conferences. There is also significant interest from marine scientists, economists, resource managers, political scientists, international lawyers, geographers and anthropologists. All these journals, conferences and marine specialists are closely focusing on (i) fabrication, launching, installation and decommissioning techniques [6] [7], (ii) fatigue and fracture [8] [9] [10], (iii) seabed foundations and structural interaction [11], (iv) Subsea engineering [12] [2], (v) Hydrodynamics and propulsion [13], and so on.

Among these research studies, we focus on Subsea Pipeline Installation (SPI) process, which has received much attention in the literature [14] [15]. The authors of [16] [17] [18] discussed S-Lay and J-lay's techniques for SPI process to define the vessel capability. These two techniques result in two different versions of SPI process each of which is described by its context of use. In fact, there are different contextual parameter values for the installation equipment, the required top tension, and the critical area when using versions based on S-Lay and J-Lay techniques.

The authors of [19] described different types of vessels used in the SPI process where each vessel is used in a specific context. For instance, in bad weather, companies must execute an SPI process version which uses a vessel equipped with a dynamic positioning system (DP system) to improve the control and the handling over of vessels at sea.

Regarding the version management, many contributions have recommended a version-based approach to deal with the process versions (*e.g.*, [20] [21] [22] [23] [24] [25]). [24] is a relevant contribution that has been made to deal with the variability of the SPI process. The authors of [24] introduced a comprehensive meta-model to define the core (basic) concepts for version modeling, while taking into account the main five dimensions to have a comprehensive view of processes (process, functional, operational, organizational and informational). However, these authors did not introduce the notion of context for versions of process in order to feature them and ease their reuse. In addition, and to the best of our knowledge, none of the above works has proposed tools that help companies in managing and retrieving versions of processes in a comprehensive framework.

Overall, the aforementioned studies provide companies with technical solutions for the installation of subsea pipelines. These solutions discuss different ways (*i.e.*, versions) of executing the SPI process based on the context but they do not address version management. In this paper, we defend the idea that the managing and the retrieving of versions must be addressed in a framework and this point is also highlighted by the authors of [24] [26] [27] who noticed a lack of knowledge regarding the best practices in SPI business processes. They emphasized that Business Process Management (BPM) must be integrated in maritime and marine fields. Highly motivated by BPM advantages, we aim at proposing a software tool that enables companies (i) to manage versions of SPI process, (ii) to explicitly and semantically describe each version according to its context of use, and (iii) to retrieve the appropriate version of SPI to be executed for a given context.

As each version is required in a specific context, it becomes crucial to consider the context to choose the appropriate process versions. In the BPM field, the notion of context is defined as "the minimum of parameters containing all the relevant information that impacts the design and the execution of a process" [28]. The authors of [29] have reported a comparative study of context modeling approaches based on a set of requirements, such as completeness, certainty (non ambiguity), level of formality, reasoning capabilities, and applicability to existing environments. They have also concluded that the ontology-based approach is the most promising way for both context modeling and querying.

In fact, several contributions have been introduced for context modeling. The authors of [30] proposed an approach for a Context-Aware Business Process Management that contemplates the context in the lifecycle of processes within an organization. In this sense, the context associated with the processes should be discovered, modeled, gathered, and used, monitored and maintained in a continuous way.

Moreover, several taxonomies have been proposed in order to classify these parameters of context (*e.g.*, [28] [31] [32]). We outline the largest one provided in [28] which distinguishes four types of context (i) immediate context, which covers parameters on process components, namely context of activities, events, and resources, (ii) internal context, which includes parameters on the internal environment of an organization that impacts the process, (iii) external context, which encompasses parameters related to external stakeholders of organizations, and finally (iv) environmental context, which contains parameters related to external factors. Finally, the authors of [33] [34] proposed ontologies for the modeling of the context in order to ensure semantic interoperability. However, the considered works poorly took advantage of ontology as their authors did not implement any reasoning strategy as no rules were modeled. Finally, these works did not address process querying using the ontology.

As for the retrieval of the process versions, a recent survey has demonstrated the lack of, and the need for, a dedicated precise contextual-based process querying [35]. This conclusion has also been confirmed more recently in [36]. Indeed, most contributions are about structural and behavioral query and there is a missing effort for contextual querying.

It is for this reason that it becomes a challenge for us to offer a new software tool that assists the BPM practitioners (users) to address the complexity of managing and retrieving process versions using both the version-based approach and the ontology-based approach.

III. Modeling versions of the SPI process

The SPI process is a collective process involving several partners. The first is, SAROST [37], which is a Tunisian company providing specialized and integrated services in the field of water and energy transportation in a sub aquatic environment where it is committed to the installation of subsea pipelines. The other partners are a Client company (Client), asking SAROST for the installation of a subsea pipelines and a Bureau Veritas (BV) to which the certification of the pipeline can be assigned. A simplified view of SPI is provided below.

We have modeled several versions of this collective process (*i.e.,* a process that involves several partners) and of its components, namely several versions of the involved partners' individual processes (*i.e.,* individual process of SAROST, individual process of Client and individual process of BV) and several versions of public or private activities. Private activities are internal activities, *i.e.* activities that describe the know-how of companies while public activities are external activities, *i.e.* activities that support interaction between companies' processes by sending or receiving messages. Table 1 summarizes the modeled versions. We also show how to specify the explicit context and the deduced context.

Versions	**Number of versions**
Collective process (SPI)	12
SAROST Individual process (SAR)	12
Client Individual process (Client)	2
BV Individual process (BV)	2
SAROST Assembly activity (Ass)	10
SAROST Control activity (Control)	2
SAROST Lay activity (Lay)	7
SAROST Receive order activity (RO)	1
SAROST Specify team activity (ST)	1
SAROST Test campaign activity (TC)	1
SAROST Prepare certificate activity (ST)	1
SAROST Send certificate activity (SC)	1
SAROST Send request for certification activity (SRC)	1
SAROST Receive certificate activity (RC)	1
SAROST Prepare PV activity (PPV)	1
SAROST Send PV activity (SPV)	1
Client Send order activity (SO)	1
Client Receive certificate activity (RC)	1
Client Receive PV activity (RPV)	1
Client Send request for certification (SRC)	1
BV Receive request for certification (RRC)	2
BV Send certificate (SC)	2

Table 1. Modeled Versions for SPI

Due to lack of space, we explain below only 3 versions of SPI collective process (among the twelve modeled). In the first version, the process is initiated by a client needing to install a new subsea pipeline. SAROST is solicited for this installation. First, the Client sends an installation order to the SAROST Company. Once the order is received, SAROST specifies the necessary team and equipment, then, it proceeds to assemble and control the pipes on shore by welders, pipefitters and controllers. The next activity is the laying of pipes offshore by the divers. Finally, when the installation is over, a test campaign has to be performed. It should be noted that the assembling, control and laying have to be repeated until reaching the pipeline length. After the test campaign, SAROST prepares an acceptance certificate and sends it to the Client. Figure 1(a) shows the SAROST's view of this first version of SPI. In fact, we only report on SAROST's private and public activities along with public activities of individual processes of the other involved partners. According to SAROST's domain experts, this first version of SPI process is defined in the following context: sea depth is less than 50 meters, pipeline length is less than 10 km and transported substance is water.

When oil or gas is transported in their pipeline with a length that can reach 50 km, a new version of SPI is defined. In this second version, SAROST subcontracts the test campaign and certification to an external company; the Bureau Veritas company (BV). Thus three partners are involved in this process version: (i) the Client, who is still the initiator of the process, (ii) SAROST to which the Client sends a request for an installation order, and (iii) BV, which performs the test of the whole pipeline.

Figure 1(b) shows the schema of this second version of SPI. In the case of a big project involving the installation of long and deep-water pipelines, the Client is responsible for contracting BV instead of SAROST. Thus SAROST proceeds only to the pipeline installation and sends back a report to the Client. Figure 1(c) shows the SAROST's view of this third version of SPI.

In order to explain why it is necessary to model so many versions, we have to feature in which context these versions must be used. Furthermore, we consider versions of the activities, such Assembly, Control and Lay, as numerous versions that have been modeled for them. According to SAROST experts, the Assembly activity depends upon the following contextual parameters:

- *Transported substance*, which can be *oil*, *gas* or *water*. This is an important parameter as the material used for the pipeline is derived from it. Indeed, according to domain experts, if the substance is water, then the material for the pipeline can be plastic HDPE, while, if the transported substance is oil or gas then, the material of the pipeline must be steel.
- *Installation technique*, which can be *floating* or *sliding*. The floating technique means that the assembled sections of the pipeline float on the water surface before landing in the seabed. On the other hand, the sliding technique means that the assembled sections of the pipeline are dragged on the seabed as the new sections are assembled to the pipeline already laid.
- *Sea depth*, which can be less than 50 meters or longer than 50 meters. Indeed, the depth affects both the divers' skills and the diving methods. If the sea depth is less than 50 meters then, at least class-2 divers must dive for assembling, while, if it is more than 50 meters, at least class-3 divers must dive. Regarding the diving method,

diving with umbilical has to be used when depth is less than 50 meters, while diving with a wet bell has to be used when depth is more than 50 meters.

- *Pipeline length*, which can be less than 10 km, between 10 and 50 km, or over 50 km. This length affects the number of the welders and/or pipefitters involved in the assembling activity. Indeed, if the length is less than 10 km, then, 6 to 8 welders and 6 to 8 pipefitters are needed in the case where the transported substance is oil or gas, while only 6 to 8 pipefitters are needed if the transported substance is water. In addition, beyond 10 km of length, the transported substance can be only oil or gas. If the length is between 10 and 50 km, the assembly activity requires more than 8 welders and 8 pipefitters, whereas if the length is over 50 km, the assembly activity requires more than 14 welders and 14 pipefitters.

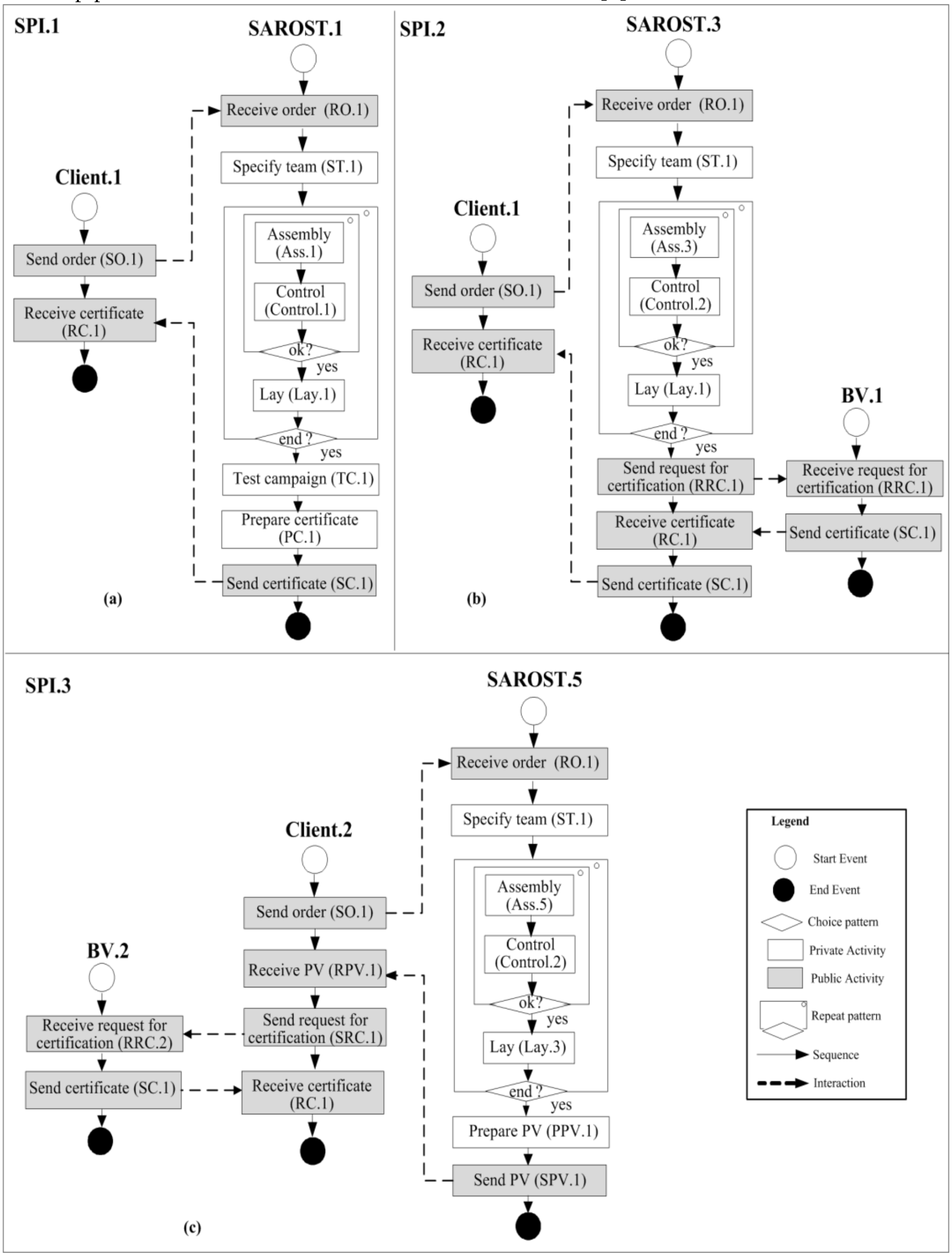

Figure 1. Three Versions of the Collective Process SPI

On the basis of the previous description, we can distinguish two types of context: (i) an Explicit Context, which is described by the parameters *Sea depth*, *Pipeline length*, *Installation technique* and *Transported substance*, and (ii) a Deduced Context, which is derived from the explicit one according to the **business rules** identified by SAROTS's experts. Table 2 shows the different versions of the Assembly activity and their corresponding context, where each of which is based on a different combination of the context parameter values. These parameters are connected to one another by the logical connector *"and"*. It should be noted that *Ass.1*, *Ass.3* and *Ass.5* are the versions involved in the 3 presented versions of SPI (*cf. Figure1).*

Any assembly activity undergoes a checking to ensure the waterproofing of the welded sections. This is the aim of the Control activity for which we have defined two versions. The first is *Control.1*, which is used when the transported substance is water (the control is simple). The second is *Control.2*, which is used when the transported substance is oil or gas. In this case, the control activity is more complex and requires providing a video proving the waterproofing of the pipeline. Regarding the Lay activity, the Sea depth and the Pipeline length have to be taken into account. Table 3 gives the context of the different versions of the Lay activity. It should be noted that these contextual parameters are connected to one another by the logical connector *"and"*. In the Assembly activity, we distinguish the Explicit Context, for the Lay activity, including *Sea depth* and *Pipeline length* parameters and its Deduced Context, which includes the parameters; *Diver skill*, *Diving method* and *Number of divers*. Note that *Control.1*, *Control.2*, *Lay.1* and *Lay.3* are the versions involved in the 3 presented versions of SPI (*cf.* Figure 1).

Regarding the other activities of the SAROST company, only one version has been modeled as no contextual parameters have been defined (*e.g.*, Receive order, Specify team, Test campaign). As a consequence, only the contexts of the Assembly, Control and Lay activities have to be considered to define the context of the SAROST's Individual Process versions. The explicit and deduced context parameters are those identified for the Assembly, the Control and the Lay activities and the combination of these parameters led us to derive 12 versions for the SAROST's individual process. Table 4 gives the context of the different versions of this individual process. We should recall that *SAROST.1*, *SAROST.3* and *SAROST.5* are the versions involved in the 3 presented versions of SPI (*cf.* Figure 1).

Assembly version number	**Explicit Context**				**Deduced Context**				
	Sea depth	Pipeline length	Installation technique	Transported substance	Pipeline material	Diver skill	Diving method	Number of welders	Number of pipefitters
Ass.1	<50	<10	floating	water	HDPE	class-2	umbilical	0	6 to 8
Ass.2	<50	<10	sliding	water	HDPE	class-2	umbilical	0	6 to 8
Ass.3	<50	<10	floating	gas or oil	steel	class-2	umbilical	6 to 8	6 to 8
Ass.4	<50	10 to 50	floating	gas or oil	steel	class-2	umbilical	>8	>8
Ass.5	<50	>50	floating	gas or oil	steel	class-2	umbilical	>14	>14
Ass.6	>=50	10 to 50	sliding	gas or oil	steel	class-3	wet bell	>8	>8
Ass.7	>=50	>50	sliding	gas or oil	steel	class-3	wet bell	>14	>14
Ass.8	<50	<10	sliding	gas or oil	steel	class-3	wet bell	6 to 8	6 to 8
Ass.9	<50	10 to 50	sliding	gas or oil	steel	class-3	wet bell	>8	>8
Ass.10	<50	>50	sliding	gas or oil	steel	class-3	wet bell	>14	>14

Table 2. Context for Assembly versions

Lay version number	**Explicit Context**		**Deduced Context**		
	Sea depth	Pipeline length	Diver skill	Diving method	Number of divers
Lay.1	<50	<10	class-2	umbilical	6 to 8
Lay.2	<50	10 to 50	class-2	umbilical	>8
Lay.3	<50	>50	class-2	umbilical	>14
Lay.4	50 to 90	10 to 50	class-3	wet bell	>8
Lay.5	50 to 90	>50	class-3	wet bell	>14
Lay.6	>90	10 to 50	class-3	saturation	>8
Lay.7	>90	>50	class-3	saturation	>14

Table 3. Context for Assembly versions

SAROST version number	**Explicit Context**				**Deduced Context**					
	Sea depth	Pipeline length	Instal-lation technique	Trans-ported substance	Pipeline material	Diver skill	Diving method	Number of welders	Number of pipefitters	Number of divers
SAROST.1	<50	<10	floating	water	HDPE	class-2	umbilical	0	6 to 8	6 to 8
SAROST.2	<50	<10	sliding	water	HDPE	class-2	umbilical	0	6 to 8	6 to 8
SAROST.3	<50	<10	floating	gas or oil	steel	class-2	umbilical	6 to 8	6 to 8	6 to 8
SAROST.4	<50	10 to 50	floating	gas or oil	steel	class-2	umbilical	>8	>8	>8

SAROST.5	<50	>50	floating	gas or oil	steel	class-2	umbilical	>14	>14	>14
SAROST.6	>=50	10 to 50	sliding	gas or oil	steel	class-3	wet bell	>8	>8	>8
SAROST.7	>90	10 to 50	sliding	gas or oil	steel	class-3	saturation	>8	>8	>8
SAROST.8	50 to 90	>50	sliding	gas or oil	steel	class-3	wet bell	>14	>14	>14
SAROST.9	>90	>50	sliding	gas or oil	steel	class-3	saturation	>14	>14	>14
SAROST.10	<50	<10	sliding	gas or oil	steel	class-2	umbilical	6 to 8	6 to 8	6 to 8
SAROST.11	<50	10 to 50	sliding	gas or oil	steel	class-2	umbilical	>8	>8	>8
SAROST.12	<50	>50	sliding	gas or oil	steel	class-2	umbilical	>14	>14	>14

Table 4. Context of SAROST's individual process versions

Finally, we can deduce the context for the 12 versions of the SPI collective process from the context of their individual involved processes. More precisely, the explicit and deduced contextual parameters are those identified in the SAROST's individual process context. Table 5 gives the context of the 3 presented versions of SPI (*cf.* Figure 1).

SPI version number	**Explicit Context**					**Deduced Context**				
	Sea depth	Pipeline length	Installation technique	Transported substance	Pipeline material	Diver skill	Diving method	Number of welders	Number of pipefitters	Number of divers
SPI.1	<50	<10	floating or sliding	water	HDPE	class-2	umbilical	0	6 to 8	6 to 8
SPI.2	<50	<10	floating	gas or oil	steel	class-2	umbilical	6 to 8	6 to 8	6 to 8
SPI.3	<50	<10	floating	gas or oil	steel	class-2	umbilical	>14	>14	>14

Table 5. Context of SPI's collective process versions

IV. Version Management and Retrieval Tool

Given the high number of versions of the SPI process, we introduce, in this section, a software tool which assists SAROST and its stakeholders (*i.e.*, Client and BV) in managing and retrieving adequate versions to be executed according to a specific context. This tool is called VMR (which stands for Version Management and Retrieval). Figure 2 provides an overview for the VMR tool architecture.

This architecture is structured in three layers: *Presentation*, *Operation* and *Persistence*. The presentation layer contains two interfaces: Version Management Interface, which supports the definition and management of process versions, and Version Retrieval Interface, which supports the retrieval of process version according to a given context. We explain respectively in the two following sub sections the operation and the persistence layers.

A. The Operation layer

The Operation layer contains the following modules:

- *Create/Update/Delete Versions of activities:* to create, update or delete the versions of activities and save them on the Version repository.
- *Create/Update/Delete Versions of processes:* to create, update or delete the versions of processes and save them on the Version repository. A complete video demonstrating the different steps of the Version Management Interface using our tool available at: https://www.youtube.com/watch?v=gNOVdznxNdU.

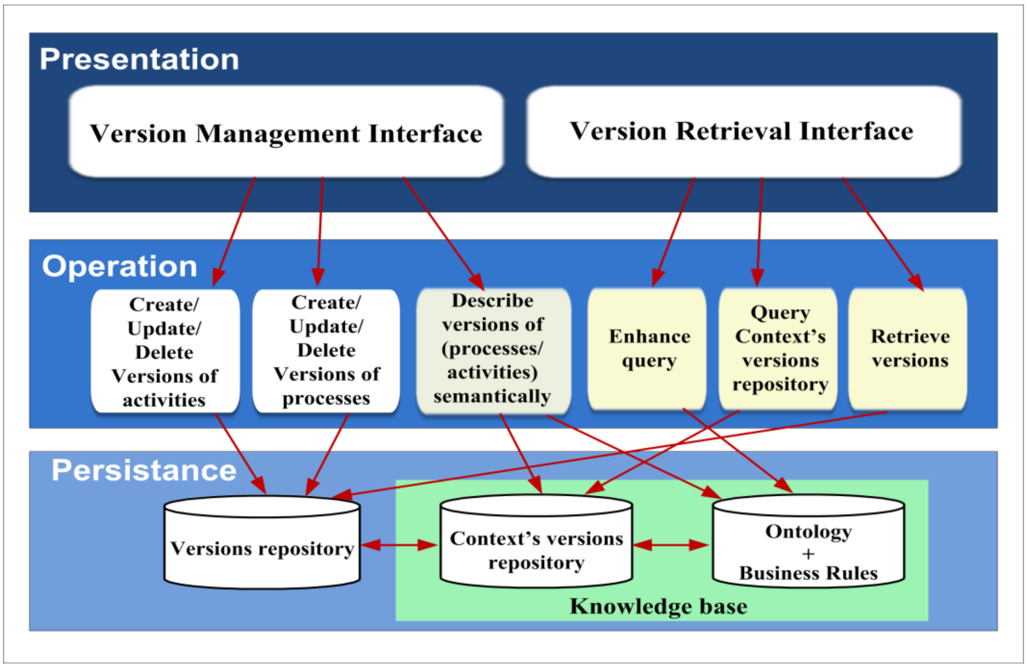

Figure 2. VMR tool architecture

• *Describe versions of (processes/activities) semantically:* to describe the context of each version. This component integrates for each version its semantic aspects which are based on an ontology and business rules. An ontology is a specification of a conceptualization [38]. It is suitable to clearly and explicitly describe the context parameters [29]. Ontologies are very used in the domain of semantic web [39] for many reasons: First, they enable knowledge sharing between systems. Second, they allow efficient reasoning on context parameter using business rules. Finally, they enable semantic interoperability between the involved companies, and their software systems. Thus we propose an ontology called Context-Onto-SPI that contains all the context parameters necessary for the description of versions of the SPI process along with the corresponding business rules.

Figure 3 shows a partial representation of the proposed ontology. Specific context parameters of SPI process are shown in grey while white ones describe context parameters independently from any process. The full copy of this ontology is available online at: https://github.com/Onto-VP2M/context-BPM/blob/master /myOnto.owl.

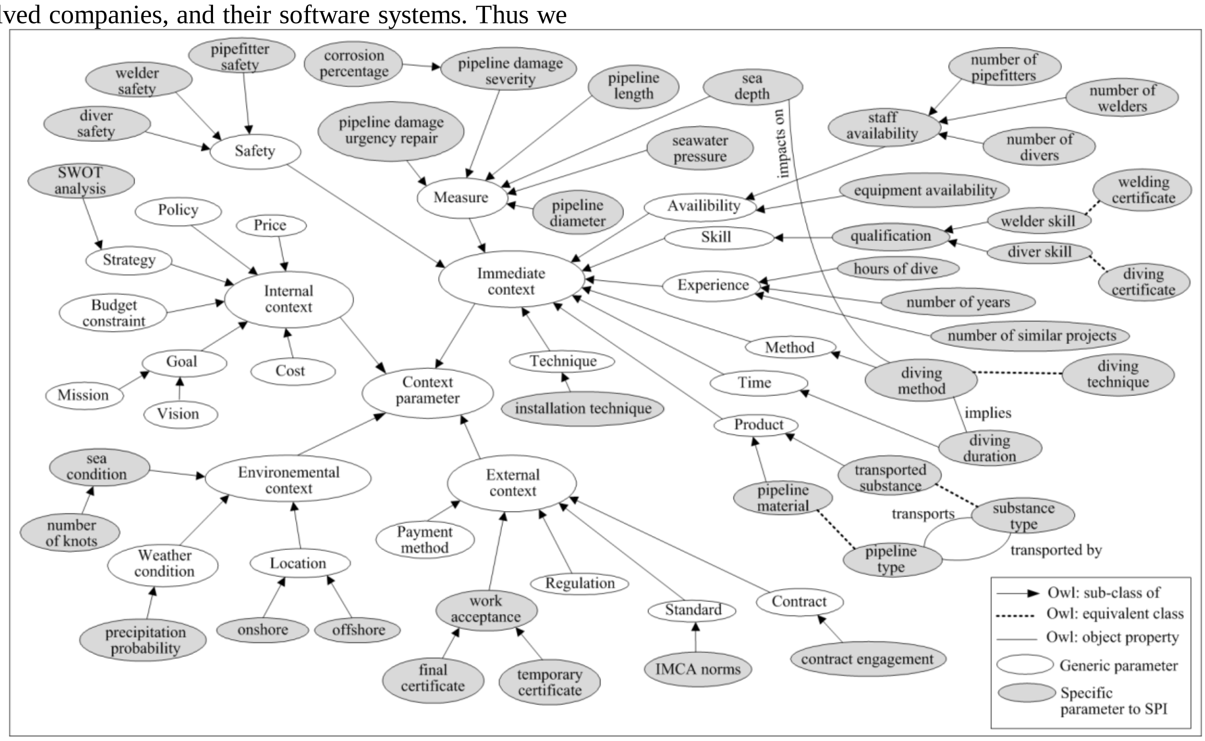

Figure 3. Context-Onto-SPI

Figure 4 shows four child interfaces (❶, ❷, ❸, ❹) of the Version Management Interface. These interfaces describe the creation of a new version of activity, a new version of individual process, a new version of collective process and a description of a process version context.

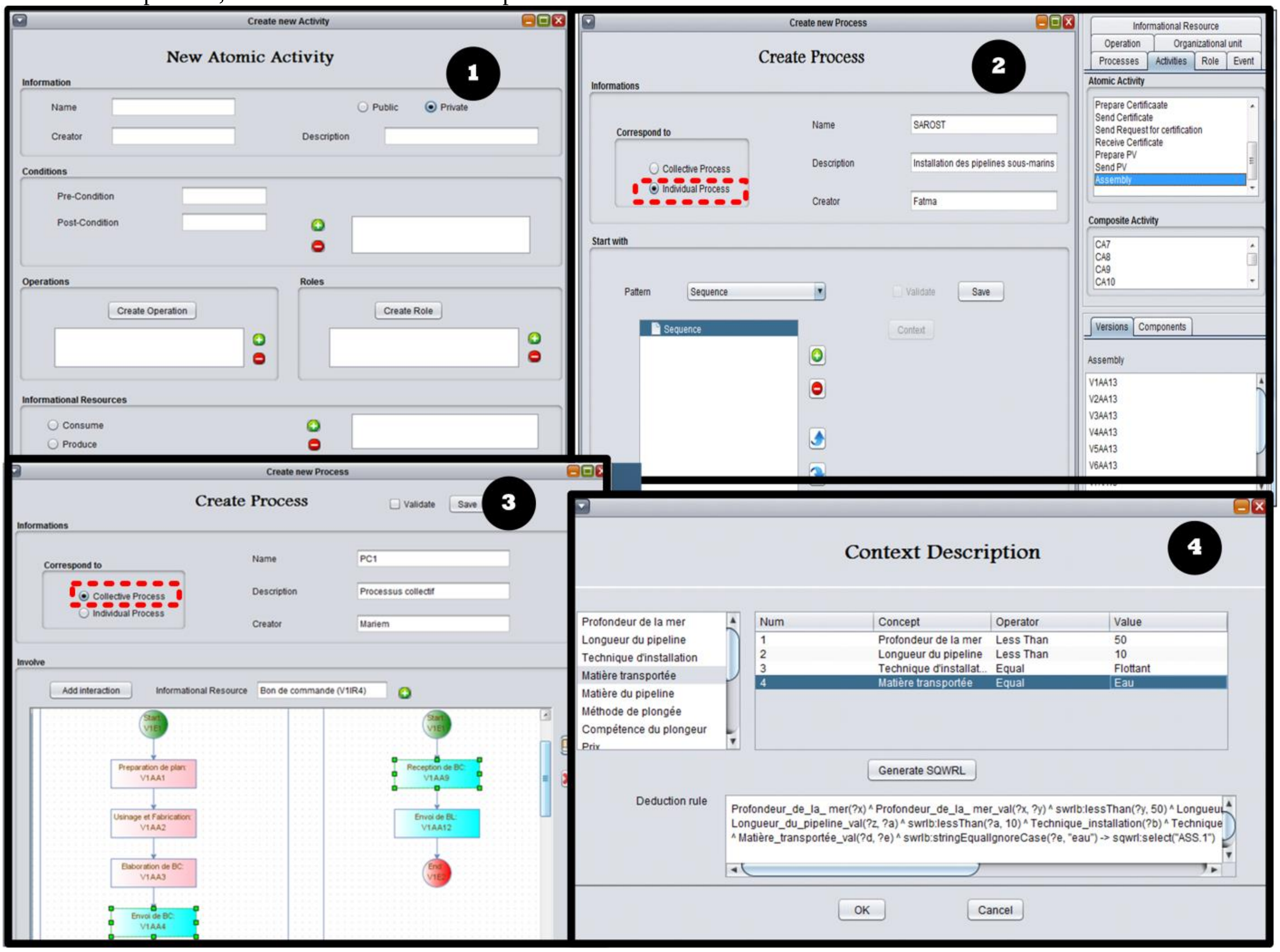

Figure 4. Managing versions interfaces

• *Enhance query*: queries come from companies or their stakeholders in order to retrieve the appropriate version of the SPI process to be executed according to a specified context. For the sake of simplification, we refer to companies and stakeholders by “User”. Four types of a query can be proposed:

- TQ#1: a complete query that contains all the parameters of the Explicit Context (cf. Tables 2, 3, 4 and 5).
- TQ#2: an incomplete query that contains a few parameters of the Explicit Context and/or of the Deduced Context.
- TQ#3: a query that contains misnomers, *e.g.*, the user writes “*sae depht*” instead of *sea depth*.
- TQ#4: a query that contains synonyms or antonyms for parameters of the Explicit Context and/or the Deduced Context.

For the TQ#2, TQ#3 and TQ#4, the component “Enhance query” rewrites the query to improve the chances of retrieving the most appropriate version according to the context described in the query. The “Enhance query” takes advantages of the ontology and business rules to enhance and enrich the query by resolving the semantic problems.

• *Query Context versions repository*: to query the repository containing the context of the versions.

• *Retrieve versions*: Extract the whole version of a process or an activity from the version repository. Figure 5 provides an overview of this component

B. *The Persistence layer*

The persistence layer provides a storage of versions of the processes/activities and their contexts. It should be noted that the Context version repository and the business rules constitute a knowledge base.

We have conducted a set of interviews with SAROST domain experts to collect business rules. As shown in Tables 2, 3, 4 and 5, business rules enhance the context of a version by a deduced context. For example, we give the following three examples of business rules written with an if-then statement:

BR1: **If** pipeline length < 5 **then** transported substance = "water"

BR2: **If** transported substance = "water" **then** pipeline material = "HDPE"

BR3: **If** pipeline material = "HDPE" **then** installation technique = "floating" and transported substance = "water".

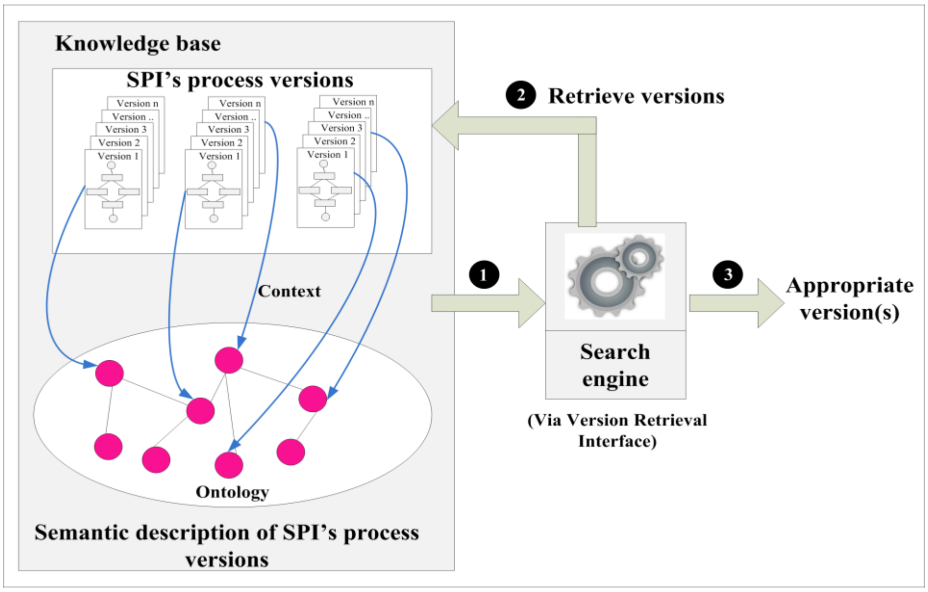

Figure 5. Overview of the component *"Retrieve versions"*

These business rules are used in two phases:

- When describing the context of the versions: SAROST is responsible for the description of the Explicit Context, however the deduced one is obtained through the business rules.
- When querying versions using their context: SAROST or its stakeholders are responsible for the description of a query containing explicit context parameters and/or deduced context parameters as they are not aware of this difference. The left part of Figure 6 shows our interface for the retrieval versions based on their semantic description. In this example, we suppose that a user indicates that depth is equal to 9 meters and length is equal to 4 kilometers. Each of these conditions is defined as a row of the grid. Thus, the specified query indicates that the user is interested in finding process versions of the SAROST process appropriate for a 4-kilometer long subsea pipeline installation at a sea depth of 9 meters. Note that all the rows specified in the grid are conditions correlated by "AND". However, it is possible to modify this correlation which indicates the logical connectors to be used in the *Logical expression for context condition* area. Each condition has a condition number which is used to define the logical expression (or and parenthesis can be used). The aim of the *Check* button is to verify whether the specified expression is correct or not. Finally, to submit a query, the user has to click on the Submit button.

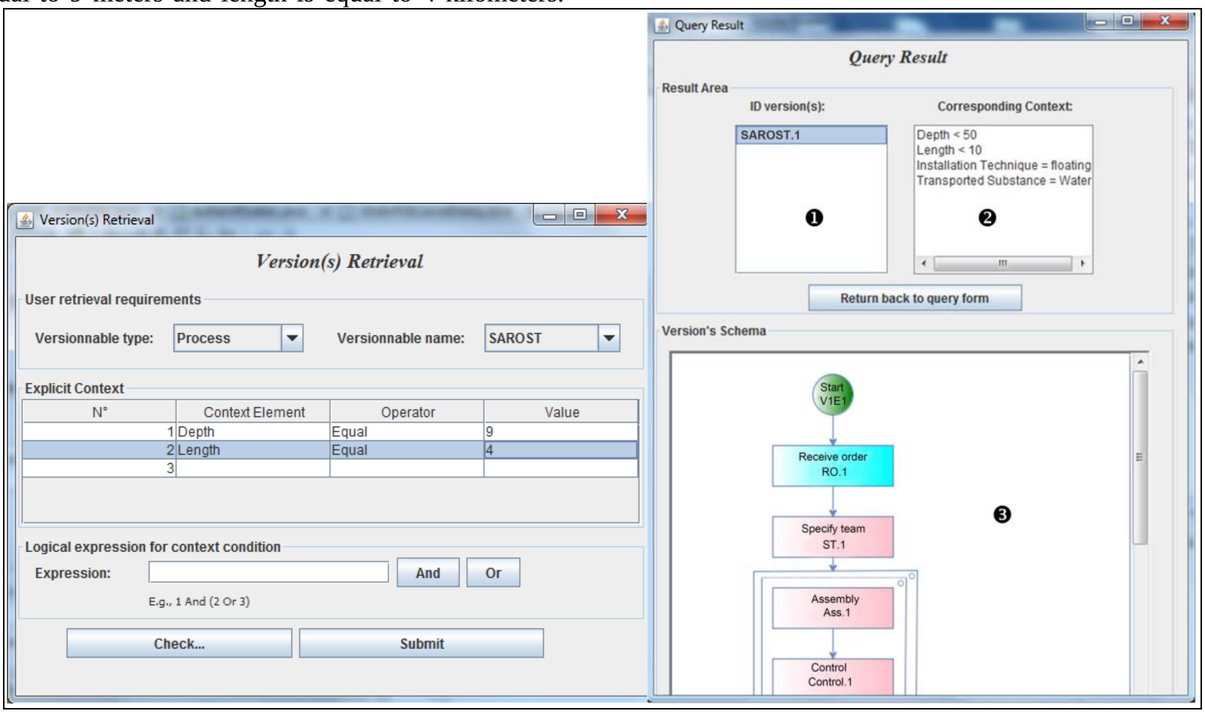

Figure 6. Example of query

Once the Submit button is clicked, the component "Enhance query" is activated to enhance the submitted query (cf. Figure 2). In the following sub-section, we will explain in detail the retrieval steps and how semantic aspects can help to retrieve the most appropriate versions.

C. Version Retrieval Steps

The retrieval component of the VMR software system is implemented according to the Onto-VP2M approach presented in [40]. Indeed, once the user submits the query, five major steps are possibly performed (cf. Figure. 7):

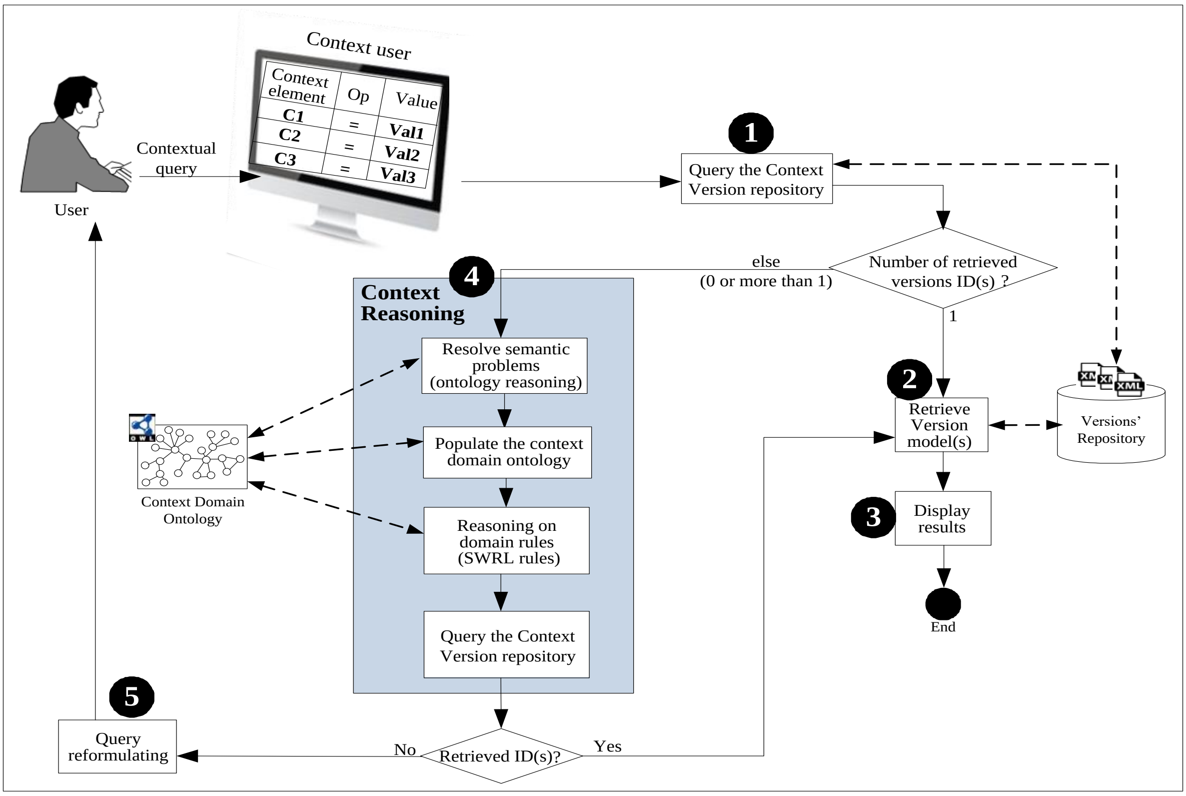

Figure 7. Query Steps

Step ❶ – Query the Context Version repository: In this step, the Context Version repository is queried and three cases may arise:

Case-1. There is exactly one version that satisfies the user's query. In this case, only steps ❷ and ❸ are performed.

Case-2. There is no retrieved version: the user may not use the same context parameters that are stored in the Context Version repository. In this case, step ❹ is performed.

Case-3. There are many retrieved versions satisfying the user's Context. In this case, step ❹ is performed.

Step ❷ – Retrieve version model(s): In this step, the process version is retrieved from the version repository.

Step ❸ – Display results: In this step, the appropriate process version (or the appropriate entity versions from Cases-2 and -3) is graphically displayed to the user.

Step ❹ – Context Reasoning: In order to enhance the query process, this step contains four modules that refer to the context domain ontology. The first module, which resolves semantic problems, is of utmost importance, especially, when there is no previously retrieved version (case 2). For example, a user may express his context using context parameters which are synonyms or misnomers to those stored in the Context Version repository. This problem can be simply resolved thanks to the equivalent class or to the SKOS annotation (*i.e.*, altLabel) of the domain context ontology. The second module populates the context domain ontology defining individuals Val1, Val2, Val3, respectively, to the corresponding context parameters C1, C2, C3 of the domain context ontology (cf. Context user in Figure 7). The third module executes the SWRL rules to deduce new knowledge that enhances the user's context. The aim of this inference is to gather a set of context parameters helping the retrieval of appropriate versions. Finally, the fourth module queries the Context Version repository. If one or more ID is returned, then steps ❷ and ❸ are performed. However, if there is no returned ID, step ❺ is performed.

Step ❺ – Query reformulating: This step notifies the user through a message indicating that there is no version that satisfies his requirement and requests him to formulate a new query.

It should be noted that the reason for querying the Context Version repository in step 1 aims at reducing the query execution time, especially, when there is no need for the context reasoning (case 1).

Figure 8 shows how the submitted query shown in Figure 6 is enhanced thanks to the first, the second and the third modules. More precisely, the first module checks the semantic of the query, and then deduces that *Depth* corresponds to *Sea depth* and *Length* corresponds to *Pipeline length*. It should be recalled that *Sea depth* and *Pipeline length* are concepts of the ontology Context-Onto-SPI (cf. Figure 3). Secondly, the second module creates individuals for "Sea depth" and "Pipeline length" context parameters having respectively 9 and 4 as values (cf. Figure 8). After an inference mechanism (the third module), the query is enhanced thanks to the SWRL rules of the context domain ontology R1, R2 and R3, which are defined above. Once the SWRL inference is made, the domain context ontology is enhanced by new individuals (*e.g.*, the execution of the above SWRL R1 infers "water", which is a new individual for "transported substance"). Figure 8 shows the new rewritten query in the dotted blue box. According to the context described in this new query, only the version identified by SAROST.1 is retrieved (cf. Table 4). The right part of Figure 6 shows the result of this query. Once the "SAROST.1" ID is selected (❶), a comprehensive description of the corresponding context is visualized in❷, and a schema of the corresponding version is visualized in❸. Subsequently, it is clear that the Context Reasoning improves the retrieval mechanism since there is no retrieved version for that query example without it.

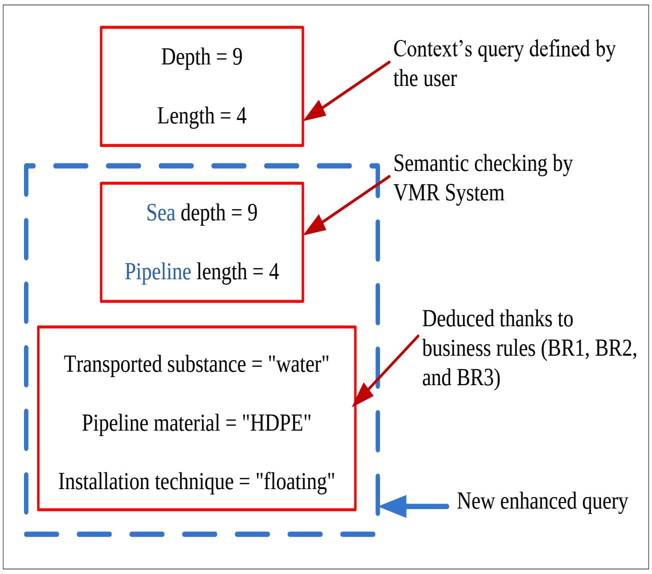

Figure 8. Example of an enhanced query

V. Evaluation

The VMR tool can be evaluated according to two criteria: (i) a quantitative evaluation that measures the tool performances and (ii) a qualitative evaluation showing the benefits and capabilities of the tool.

In the quantitative evaluation, we consider the types of queries already mentioned in section B (cf. "Enhance query" component). For each of these types, we compare the obtained results of the VMR software tool using semantic aspects (*i.e.*, ontology) to the results with an existing software tool used by SAROST experts. Note that the SAROST's tool does not use an ontology and thus does not take advantage of the reasoning capabilities of ontologies.

We have measured the recall and precision metrics for a set of assessment queries belonging to all these different types related to the SPI process. We present formulas for the Precision, Recall and F-measure metrics hereinafter in (1), (2) and (3). These formulas are expressed in terms of (i) query true positive (TP), *i.e.*, number of retrieved versions that are relevant, (ii) query false positive (FP), *i.e.*, number of retrieved versions that are irrelevant, and (iii) query false negative (FN), *i.e.*, number of relevant versions that are not retrieved. Thus, (1) the precision metrics refers to the number of the retrieved versions that are relevant, divided by the total number of the retrieved versions, (2) the recall metrics refers to the number of the retrieved versions that are relevant, divided by the number of the relevant versions that should have been retrieved, (3) while the F-measure metrics refers to the balanced mean between the precision and recall metrics. The more F-measure approaches 1, the more retrieval is efficient.

$$Precision = \frac{TP}{TP+FP} = \frac{Number\ of\ Relevant\ Versions\ Retrieved}{Number\ of\ Versions\ Retrieved} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} = \frac{Number\ of\ Relevant\ Versions\ Retrieved}{Number\ of\ Relevant\ Versions} \quad (2)$$

$$F-measure = 2*\frac{Precision\ *\ Recall}{Precision\ +\ Recall} \quad (3)$$

TP: True Positive FP: False Positive FN: False Negative

We have considered several queries for each type, and for each of these queries, we have calculated its TP, FP and FN. Table 6 summarizes these achieved results.

	VMR software tool using semantic aspects						**SAROST's existing software tool without using semantic aspects**					
	TP	FP	FN	Precision	Recall	F-measure	TP	FP	FN	Precision	Recall	F-measure
TQ#1	20	0	0	1	1	1	20	0	0	1	1	1
TQ#2	35	24	14	0.59	0.71	0.64	25	70	31	0.26	0.44	0.32
TQ#3	22	9	13	0.7	0.62	0.65	0	0	0	0	0	0
TQ#4	26	6	9	0.81	0.74	0.77	0	0	0	0	0	0

Table 6. Values of Precision Recall and F-measure Metrics

Let us examine some of these results. First, we examine the relevant queries for TQ#1. As indicated in Table 6, the precision and recall metrics are both equal to 1 for both cases as the user's context matches exactly the Explicit Context stored in Context Versions Repository (cf. Figure 2).

Regarding the relevant queries for TQ#2, querying without semantic aspect sends back the precision and recall metrics what have an average equal to 0.26 and 0.44, respectively. On the other hand, querying with semantic aspect sends back the precision and recall metrics the values of which are on average equal to 0.59 and 0.71, respectively. As a consequence, the F-measure result for querying with semantic aspect is better than the F-measure result for querying without semantic aspect (*cf.* Figure 9). These results show the importance of using business rules. Indeed, for TQ#2, the business rules play an important role in enhancing the query by other context parameters.

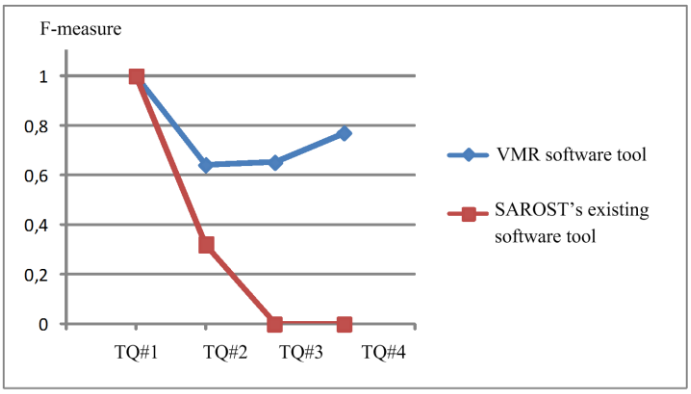

Figure 9. F-measure curve

Finally, regarding the queries illustrating TQ#3 and TQ#4, retrieval with semantic aspect is more efficient than the one without. Indeed, the VMR tool can detect the semantic problems (*e.g.*, misnomer, synonym or antonym) and resolve them thanks to ontology capabilities.

As for the qualitative evaluation, we compare our tool to two existing popular BPM tools from the literature, namely Signavio [41] and Aeneis [42] tools. Indeed, these BPM tools ensure the process versions modeling. However, the major limitation of Signavio and Aeneis is that they focus on process versions modeling without supporting the context modeling of these versions. In addition, the retrieval mechanism in these tools is very basic and is confined to the search for versions by keywords captured from text annotations, such as the process name, the publishing date, the author's name, the revision comment, etc. They fall short of providing any guideline that help to retrieve the adequate version of process appropriate to a specified context. To sum up, we recommend the VMR software tool which emphasizes the requirement of context-awareness in the process versioning (i) to structure versions of process and (ii) eventually create context-based queries to search and find adequate versions of process. For this purpose, it recommends an ontology-based approach for context modeling and querying. Indeed, it is worth taking advantage of the ontology to ensure a semantic interoperability between BPM practitioners.

VI. Conclusion

New technologies, governmental rules, organizational context, adoption of new standards and so on, lead companies to define several versions for their processes. Despite this reality in companies, the questions of managing these versions and reusing them arise. This is the case for the SAROST Company, which is left with a high number of versions for its Subsea Pipeline Installation (SPI) process as well as its corresponding activities. This paper shows how we model SPI process using a version-based approach. Then, it provides a solution to manage and retrieve (VMR) versions of the SPI process. Specifically, this paper proposes the VMR software tool that enables (i) the management of all the versions in terms of creation, updating and deletion, (ii) the integration of semantic aspects for each version to represent its context of use, and (iii) the retrieval of the most appropriate version(s) for a specific situation. The evaluation of VMR outlines the suitability for using semantic aspects, based on an appropriate ontology for managing and retrieving versions.

Although the SPI process is considered as our starting point to contribute to this field, VMR tool can be used for both other processes and other fields (*e.g.*, medical field, automotive field).

Future research will address the weaknesses of our contribution. Regarding the VMR tool, we have planned to take into account privacy aspects when querying the process versions. Moreover, we have foreseen to evaluate the VMR tool usability, i.e. VMR tool acceptance in other fields, in addition to the evaluation reported in the paper. Regarding the provided ontology, we have two main objectives. The first is to add fuzzy annotations to express the queries in a more flexible manner since the linguistic modifiers can be used (e.g., retrieve versions of the SAR process appropriate to a significant depth) [43]. The second objective is the evaluation of the power of expressing this ontology with the Bunge Wand-Weber (BWW) ontology [44], which is known as a good theoretical framework for ontology expressiveness measure.

References

[1] A. C. Palmer, R. A. King. *Subsea pipeline engineering*, PennWell Books, 2004.

[2] Y. Bai, Q. Bai (Eds.). *Subsea engineering handbook*, Gulf Professional Publishing, 2012.

[3] M. Dumas, M. La Rosa, J. Mendling, H. A. Reijers. *Fundamentals of business process management*, Heidelberg: Springer, 2nd edition, 2018.

[4] P. Harmon. "The State of Business Process Management". A BP Trends Report, 2016.

[5] M. La Rosa, W. Van Der Aalst, M. Dumas, F. Milani. "Business process variability modeling: A survey". ACM Computing Surveys (CSUR), 50(1), 2017.

[6] N. S. Philip, S. Wilde, R. Arshad, I. Washash, T. A. R. Al-Sayed. "Decommissioning Process for Subsea Pipelines". In *Proceedings of the Abu Dhabi International Petroleum Exhibition and Conference (ADIPEC)*, Abu Dhabi, Society of Petroleum Engineers, pp. 10-13, 2014.

[7] A. Abdelaah, S. M. Cresswell, G. M. E. Manzocchi, C. Thistlethwaite. "Practical Applications of Structural Analysis Support to Decommissioning". In *Proceedings of Offshore Technology Conference (OTC)*, Houston, Texas, USA, pp 1-17, 2016.

[8] B. M. Ayyub, K. A. Stambaugh, T. A. McAllister, G. F. de Souza, D. Webb. "Structural Life Expectancy of Marine Vessels: Ultimate Strength, Corrosion, Fatigue, Fracture, and Systems", *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, 1(1), 2015.

[9] K. Doshi, T. Roy, Y. S. Parihar. "Reliability based inspection planning using fracture mechanics based fatigue evaluations for ship structural details", *Marine Structures*, pp. 1-22, 2017.

[10] R. A. Khan, S. Ahmad. "Nonlinear dynamic and bilinear fatigue reliability analyses of marine risers in deep

offshore fields", *Ships and Offshore Structures*, pp. 1-10, 2017.

[11] J. Ye, D. S. Jeng, A. H. C.Chan, R. Wang, Q. C. Zhu. "3D integrated numerical model for fluid–structures–seabed interaction (FSSI): Elastic dense seabed foundation", *Ocean Engineering*, pp. 107-122, 2016.

[12] Y. Bai, Q. Bai (Eds.). *Subsea pipelines and risers*, Elsevier, 2005.

[13] H. L. Chen, X. P. Bai, C. M. Liu, W. Nie. "Hydrodynamic analysis of barge based on SESAM software [J]", *Transducer and Microsystem Technologies*, 2009.

[14] B. Guo, S. Song, A. Ghalambor, J. Chacko. "Offshore pipelines", *Elsevier*, 2005.

[15] X. Zhang, M. Duan, D. Mao, Y. Yu, J. Yu, Y. Wang. "A mathematical model of virtual simulation for deepwater installation of subsea production facilities". *Ships and Offshore Structures*, 7(2), pp. 182-195, 2017.

[16] S. Wästberg, H. Pisarski, B. Nyhus. "Guidelines for engineering critical assessments for pipeline installation methods introducing cyclic plastic strain". In *Proceedings of 23rd International Conference on Offshore Mechanics and Arctic Engineering (ASME)*, American Society of Mechanical Engineers, pp. 19-26, 2004.

[17] E. Torselletti, L. Vitali, E. Levold, K. J. Mørk. "Submarine pipeline installation JIP: Strength and deformation capacity of pipes passing over the S-lay vessel stinger", *Strain*, pp. 535-555, 2006.

[18] J. Herdiyanti. Comparisons study of S-Lay and J-Lay methods for pipeline installation in ultra deep water. *Master's thesis*, University of Stavanger, Norway, 2013.

[19] A. J. Sørensen. "A survey of dynamic positioning control systems", *Annual reviews in control*, 35(1), pp. 123-136, 2011.

[20] P. Dadam, M. Reichert. "The ADEPT project: a decade of research and development for robust and flexible process support". *International Journal of Computer Science-Research and Development*, 23(2), pp. 81-97, 2009.

[21] M. A. Chaâbane, L. Bouzguenda, R. Bouaziz. "Towards a Business Process Versions Querying and Manipulating Language". In *Proceedings of the International Conference on Information Technology and e-services (ICITES)*, Sousse, Tunisia, pp. 12–19, 2011.

[22] C. C. Ekanayake, M. La Rosa, A. H. Ter Hofstede, M. C. Fauvet. "Fragment-based version management for repositories of business process models". In *Proceedings of the International Conferences OTM Confederated On the Move to Meaningful Internet Systems (OTM)*, Hersonissos, Crete, Greece, pp. 20-37, 2011.

[23] X. Zhao, C. Liu. "Version management for business process schema evolution". *International Journal of Information Systems*, 38(8), pp. 1046-1069, 2013.

[24] F. Ellouze, M. A. Chaabane, Bouaziz, R., E. Andonoff. "Addressing inter-organisational process flexibility using versions: The VP2M approach". In *Proceedings of the International Conference on Research Challenges in Information Science (RCIS)*, Grenoble, France, pp. 1-12, 2016.

[25] I. Ben Said, M. A. Chaâbane, R. Bouaziz, E. Andonoff. "A Version-based Approach to Address Flexibility of BPMN Collaborations and Choreographies". In *Proceedings of the 16th International Conference on e-Business (ICE-B)*, Lisbon, Portugal, pp. 31-42, 2016.

[26] C. Speier, D. Mollenkopf, T. P. Stank. "The role of information integration in facilitating 21 st century supply chains: a theory-based perspective", *Transportation Journal*, pp. 21-38, 2008.

[27] R. Elbert, H. Pontow, A. Benlian. "Standardization and integration of information flows along the maritime transport chain". In *Selected Proceedings of the 13th World Conference on Transport Research (WCTR)*, pp. 1–14, 2013.

[28] M. Rosemann, J. Recker, C. Flender. Contextualisation of business processes. *Journal of Business Process Integration and Management*, 3(1), pp. 47-60, 2008.

[29] T. Strang, C. Linnhoff-Popien. "A context modelling survey", In *Workshop on Advanced Context Modelling, Reasoning and Management*, Nottingham, United Kingdom, 2004.

[30] F. M. Santoro, F. Baião, K. Revoredo, V. Tavares, "Modeling and Using Context in Business Process Management: A Research Agenda". *Journal of Modeling and Using Context*, Published by ISTE Ltd. London, UK, pp.1-20, 2017.

[31] J. Brocke, S. Zelt, T. Schmiedel. "On the role of context in business process management". *Journal of Information Management*, 36(3), pp. 486-495, 2016.

[32] P. Brézillon, "Elaboration of the Contextual Graphs representation: From a conceptual framework to an operational software". *Journal of Modeling and Using Context*, ISTE OpenScience, 2017.

[33] O. Saidani, C. Rolland, S. Nurcan. "Towards a generic context model for BPM". In *Proceedings of the 48th Hawaii International Conference on System Sciences*. Kauai, HI, USA: IEEE, pp. 4120-4129, 2015.

[34] Y. Lassoued, L. Bouzguenda, T. Mahmoud. "Context-aware business process versions management". *International Journal of e-Collaboration*, 12(3), pp. 7-33, 2016.

[35] J. Wang, T. Jin, R. K. Wong, L. Wen "Querying business process model repositories". *Journal of World Wide Web*, 17(3), pp. 427–454, 2014.

[36] A. Polyvyanyy, C. Ouyang, A. Barros, W. M. van der Aalst. "Process Querying: Enabling Business Intelligence through Query-Based Process Analytics". *Decision Support Systems*, pp. 41-56, 2017.

[37] Sarost, http://www.sarost-group.com/ (last accessed on December 20, 2017)

[38] T. Gruber. What is an Ontology?, 1993. http://www-ksl.stanford.edu/kst/ what-is-an-ontology.html.

[39] A. Maedche, S. Staab. "Ontology learning for the semantic web", *IEEE Intelligent systems*, 16(2), pp. 72-79, 2001.

[40] F. Ellouze, MA. Chaâbane, E. Andonoff, R. Bouaziz. "Onto-VP2M: A New Approach to Model and Manage Collaborative Process Versions using Contexts and Ontologies". *International Journal of e-Collaboration* (IJeC), 13(3), pp. 39-62, 2017.

[41] Signavio (2015). Benutzerhandbuch Signavio. Retrieved from https://docs.signavio.com/userguide/editor/de/index.html

[42] Aeneis (2014). Aeneis Handbuch: intellior AG. Retrieved from https://www.intellior.ag/wp-content/uploads/Handbuch_Aeneis_Anwender.pdf

[43] H. Ghorbel, A. Bahri, R. Bouaziz. "Fuzzy Ontologies Building Platform for Semantic Web: FOB Platform", In *Next Generation Search Engines: Advanced Models for Information Retrieval*, IGI Global, pp. 92-113, 2012.

[44] Y. Wand, R. Weber. "An Ontological Analysis of some Fundamental Information Systems Concepts". In *International Conference on Information Systems* (ICIS), Minneapolis, Minnesota, USA, pp. 213–226, 1988.

Author Biographies

Fatma ELLOUZE recieved her MSc in Computer Science from the Faculty of Economics and Management of the University of Sfax, Tunisia, in November 2012. Currently, she is a PhD Student in Computer Science in the same Faculty. She is a member of the Multimedia, Information systems and Advanced Computing Laboratory, since 2013. Her current research interests include Business process management, Process modeling, Context Modeling, ontologies and Information systems.

Mohamed Amine CHAÂBANE is an associate Professor at the Informatics and Quantitative Methods Department of Higher Institute of Business Administration (ISAAS) – University of Sfax-Tunisia. He is a member of MIRACL Lab. His research and teaching interests includes Business Process Management, Information Systems, and Specification languages.

Eric ANDONOFF is an Associate Professor at University of Toulouse and researcher at IRIT laboratory. His main research area is the Business Process Management area and his investigations more particularly address business process adaptation, context modelling for processes, social business processes and process for crisis management.

Rafik BOUAZIZ, is full professor on computer science at the Faculty of Economic Sciences and Management of Sfax University, Tunisia. He was the president of this University during August 2014 – December 2017, and the director of its doctoral school of economy, management and computer science during December 2011 – July 2014. His PhD has dealt with temporal data management and historical record of data in Information Systems. The subject of his accreditation to supervise research was "A contribution for the control of versioning of data and schema in advanced information systems". Currently, his main research topics of interest are temporal databases, real-time databases, information systems engineering, ontologies, data warehousing and workflows.