VPlan - Ontology for Collection of Process Verification Data

Tomasz Miksa
SBA Research
Vienna, Austria
tmiksa@sba-research.org

Ricardo Vieira
José Barateiro
INESC-ID Information Systems Group
& LNEC
Lisbon, Portugal
{rjcv, jose.barateiro}@ist.utl.pt

Andreas Rauber
Vienna University of Technology
& SBA Research
Vienna, Austria
rauber@ifs.tuwien.ac.at

ABSTRACT

The reproducibility of modern research depends on the possibility to faithfully rerun the complex and distributed data transformation processes which were executed by scientists in order to make new scientific breakthroughs. New methods and frameworks try to address this problem by collecting evidence used for verification of such experiments. However, there is still a lack of a flexible data model which would address all of the needs of these methods. This paper presents the VPlan ontology designed for the purpose of organizing and storing of data collected for verification of preserved processes. The VPlan ontology stores and links the data extracted from the preserved process. Furthermore, it includes descriptions of actions taken to collect the data, as well as provides a clear break down of requirements that lead to its collection. We demonstrate the usage of the VPlan ontology within the preservation process and describe in detail its alignment with the Verification Framework (VFramework). In order to illustrate its applicability to the eScience domain, we evaluate it on a use case from the civil engineering domain, which is an example of a typical sensor data analysis process.

1. INTRODUCTION

The preservation of entire processes and workflows has already gained the interest of the digital preservation community [18]. There are a number of research projects [3, 11] addressing the challenges of keeping processes available in the long term. They deliver tools [8] and frameworks [17] which try to address the problem of not only preserving the data which is produced at the output of the eScience experiments, but also preserving the way the results were obtained. This includes preservation of complex and very often distributed processes which captured, processed, integrated or visualised the data. Despite these advances, the problem of reproducibility of modern data-intensive science remains unsolved and is currently receiving the attention of publishers [12], funding agencies [9] and researchers themselves [4]. As a result, scientists are often required to create data management plans in which they describe the data produced by their experiments. This solves the problem partially, because the information on processes used in the experiments are still not detailed enough. Process management plans [14] complement the data management plans with information on processes, but they are still not fully implemented.

Most of these efforts focus only on the problem of preserving the experimental data and documenting the processes executed to obtain these results. However, information needed for verification and validation of the redeployed process must also be captured. The verification of redeployed processes is a complex task and depends on many things: the way the processes are specified, the drivers for their preservation, the preservation strategies applied; the reasons for the redeployment, the redeployment environments, and so on. Such information must be collected at the time of process execution and is later used to prove that the process running in the redeployment environment is performing in the way it was originally meant. This may be crucial in litigation cases when the correctness of the original process executed at some time in the past could be questioned and the only way to check this is to re-run the original process. The verification can only be reliable when the requirements used for the verification are well structured and the processes of data capturing and redeployment quality metrics calculation are clearly defined.

In [13] we presented the VFramework which defines a framework for verification of preserved and redeployed processes. In this paper we present the VPlan which is an ontology for collection of process verification data. The VPlan stores the information collected during application of the VFramework. It integrates well with the TIMBUS Context Model [2, 11] and makes use of the ArchiMate [20] modelling language to describe the data capture processes. It also links the significant properties and metrics, which are used for verification, to the real location of data. In this paper we also demonstrate the applicability of the VPlan to the verification of preserved and redeployed eScience processes. We use a use case from the civil engineering domain which is an example of a typical sensor data analysis process.

The paper is organized as follows. Section 2 presents the state of the art. In Section 3 the VPlan is described and

mapping to the VFramework is provided. Section 4 describes usage of the VPlan in the eScience use case. We provide conclusions and future work in Section 5.

2. STATE OF THE ART

This section discusses the most important work related to the verification and validation of preserved processes. We also place this work in the context of the TIMBUS Preservation Process and explain concepts that impacted the design of the VPlan.

2.1 Verification framework

In [6] a conceptual framework for evaluation of emulation results was presented. It was demonstrated in [5] that the framework can be successfully applied to evaluate the conformance and performance quality of applications and simple processes redeployed in an emulator. This was demonstrated in case studies in which the framework was used to evaluate the emulation of a video game and an accounting program. The VFramework presented in [13] is a refinement of that framework for complex, potentially distributed processes. It provides detailed specification of actions which have to be performed for verification of redeployed processes. The VFramework is presented in Figure 1 and consists of two sequences of actions. "The first one (depicted in blue) is performed in the original environment. The result obtained from the execution of each step is written into the VPlan. The second sequence (depicted in green) is performed in the redeployment environment. The necessary information for completion of each of the steps is read from the VPlan." [13] By original environment we mean a system in which the process is executed. The redeployment environment is the system to which the process will be moved when a decision to rerun the preserved process is taken. The redeployment can take place at any time in the future when the original platform is not available anymore. Hence, it may be necessary to re-engineer the process in order to fit it into a new system.

2.2 TIMBUS Preservation Process

In [18] the TIMBUS Preservation Process for preservation of processes is presented and applied to an eScience process. The authors explain three phases of the approach: plan, preserve and redeploy. The TIMBUS Preservation Process assumes that the verification data is collected during the preserve phase and used for verification of the process in the redeploy phase. The VFramework [13] provides a detailed list of steps for performing verification when executing the TIMBUS Preservation Process. The VPlan presented in this paper describes an ontology for collection of verification data. Detailed information on the TIMBUS Preservation Process can also be found in [21].

2.3 Process modelling

Processes, as organized sets of activities performed to achieve specific desired outcomes, are something that exists in all organizations and might be described and documented in many different ways. The description of a process using a set of key concepts and relations is typically known as process modelling. Modelling enables a common understanding easing the analysis of a process [1]. There are several techniques to model processes depending on the pretended analysis, such as flow charts, data flows, and role activity diagrams [1]. The most known and used technique and language to describe the flow of a business process is the Business Process Modelling Notation (BPMN) [16].

Enterprise Architecture (EA) is a coherent set of principles, methods and models to design, analyse, change and manage organizations through four main architecture domains: business, data, application and technology. However, in order to properly describe the main concepts of EA and the dependencies between domains, BPMN is insufficient [19]. Therefore EA languages emerged in order to address the existing gap. ArchiMate [7] represents the culmination of years of work in the area of EA modelling languages and frameworks and is one of the most used EA languages nowadays. It provides high-level abstract concepts divided into three tightly connected EA layers: the business layer, the application layer, and the technology layer. It is a mature language with extensive use and practice where elements and relationships are clearly defined and explained [19].

Taking into account the advantages of Archimate against the common process modelling languages, Archimate is used to model the required processes in the VPlan presented in this paper, namely the preserved process, the capture processes and, if they exist, the determinism transformation processes.

2.4 Ontologies

Provenance ontologies seem a natural candidate to be used at least as a basis for extension in order to address the requirements of the VFramework. The Open Provenance Model[1] has a corresponding OPMO[2] ontology. It describes process execution, but does not allow for definition of one's own metrics. Similarly the information contained in the Janus [15] ontology describes execution of a workflow, i.e. data exchanged between workflow elements, timestamps, and so on. This information is useful for modelling of the process instance execution, but does not provide information on the significant properties, metrics or conditions in which the capturing took place. The Wf4Ever[3] project uses the wfprov[4] ontology that is capable of storing information about the execution and the parameters of a workflow, but there is also no information on significant properties or capture processes. Furthermore, both Janus and wfprov are limited to formally specified processes like workflows. Achieving the functionality of the VPlan by linking any other ontology to the OPMO, wfprov or Janus ontologies would not be possible and may lead to semantic inconsistencies between the concepts. None of the existing ontologies is suitable to fully address the requirements of the VFramework and neither is the composition of them.

3. VPLAN

The VPlan is an ontology-based document for storing and organizing information collected during the VFramework application. The following subsections describe: its structure, integration with the Context Model and mapping to the VFramework steps.

[1]http://openprovenance.org/
[2]http://openprovenance.org/model/opmo
[3]http://www.wf4ever-project.org/
[4]http://purl.org/wf4ever/wfprov

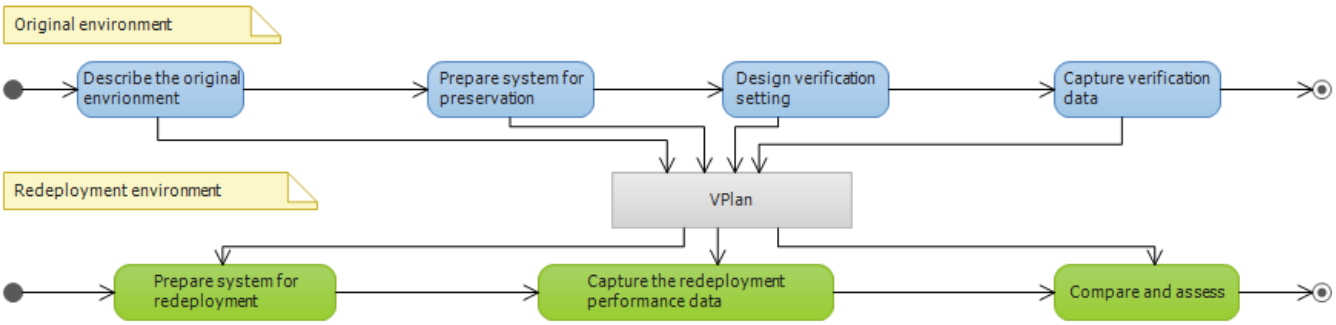

Figure 1: VFramework [13].

3.1 Overview

The VPlan[5] is created when the original process is preserved. It is accessed during the redeployment phase. The VPlan is created per process and it contains process instances which can verify particular process execution.

Figure 2 depicts the concept map of the VPlan. The names of the concepts correspond to the concepts defined in [13]. The light blue boxes are the classes, e.g. *VPlan*, *Metric*, *RedeploymentScenario*, and so on. The named arrows connecting the light blue boxes are object properties, e.g. *measures*, *appliesToScenario*, *hasInstance*, and so on. The arrows that point to the green boxes are the data properties, namely: *isLocatedAt*, *hasTextDescription* and *isInline*. There are also five dark blue boxes, which are individuals used for creating an enumeration for the *MetricTargetOperator* class. Finally, there are 3 grey boxes which depict elements imported to the VPlan by importing the TIMBUS Context Model.

In general the VPlan links the requirements expressed by significant properties and metrics with the way they are measured. To describe the measurement process, the information on process instances and capturing processes is provided. The VPlan uses the Context Model to precisely depict from which process' part the information was captured. Moreover, it includes capturing processes, which were originally modelled in ArchiMate and later converted to an ontology in order to document the way the data was collected. Finally, the VPlan stores not only information on data location used to run the process (process instances), but also the data which was captured from the process for calculation of metrics.

3.2 Relation to the Context Model

Due to the fact that the VPlan is an OWL[6] document, it benefits from integration with other ontologies. By default it is integrated with the TIMBUS Context Model. Furthermore, if different concepts are needed, the VPlan can integrate with any other existing ontology. The VPlan uses the Context Model in four different ways:

- import of the Context Model concepts at the model level,
- import of the preserved process at the instance level,
- import of the capture process at the instance level,
- import of the determinism transformation process at the instance level.

Figure 3 illustrates the relation of the VPlan to the Context Model. Each of the cases is discussed in the next subsections.

3.2.1 Import of the Context Model at the model level

The VPlan is coupled with the Context Model at the model level. This is one of the fundamental assumptions. Due to this coupling, the VPlan can make an extensive use of the machine-readable representation of the process. Moreover, the Context Model is based on the ArchiMate specification which is a recognized standard by many Enterprise Architects. Therefore, reuse of concepts from the Context Model (and indirectly from the ArichMate) in the VPlan facilitates VPlan understanding to users from these communities.

3.2.2 Import of preserved process at the instance level

The TIMBUS preservation framework assumes that in one of the initial steps a Context Model of the preserved process is created. Because the VPlan is always targeted at a particular process, then a coupling of the VPlan and the Context Model of the preserved process is natural. This is achieved by importing the ontology-based representation of the process into the instance of the VPlan. As a result, the redeployment scenarios, measurement points and levels of comparison (see [13] for definitions explanation) can easily be specified.

The redeployment scenarios can be described by connecting the *RedeploymentScenario* individual with each process step of the preserved process. As a consequence, further dependencies of each process's step can be inferred automatically without the need for explicit specification. When it comes to the specification of measurement points, they can be pointed directly to the preserved process and thus any ambiguities, which could stem from a verbal description, are removed. The levels of comparison are implicit and depend on the kind of process element to which the measurement point links.

3.2.3 Import of capture processes at the instance level

The VPlan requires that for each of the metrics a capture process is defined which describes how the data, which is

[5] http://timbus.teco.edu/svn/public/ontologies/VPlan.owl
[6] http://www.w3.org/TR/owl2-overview/

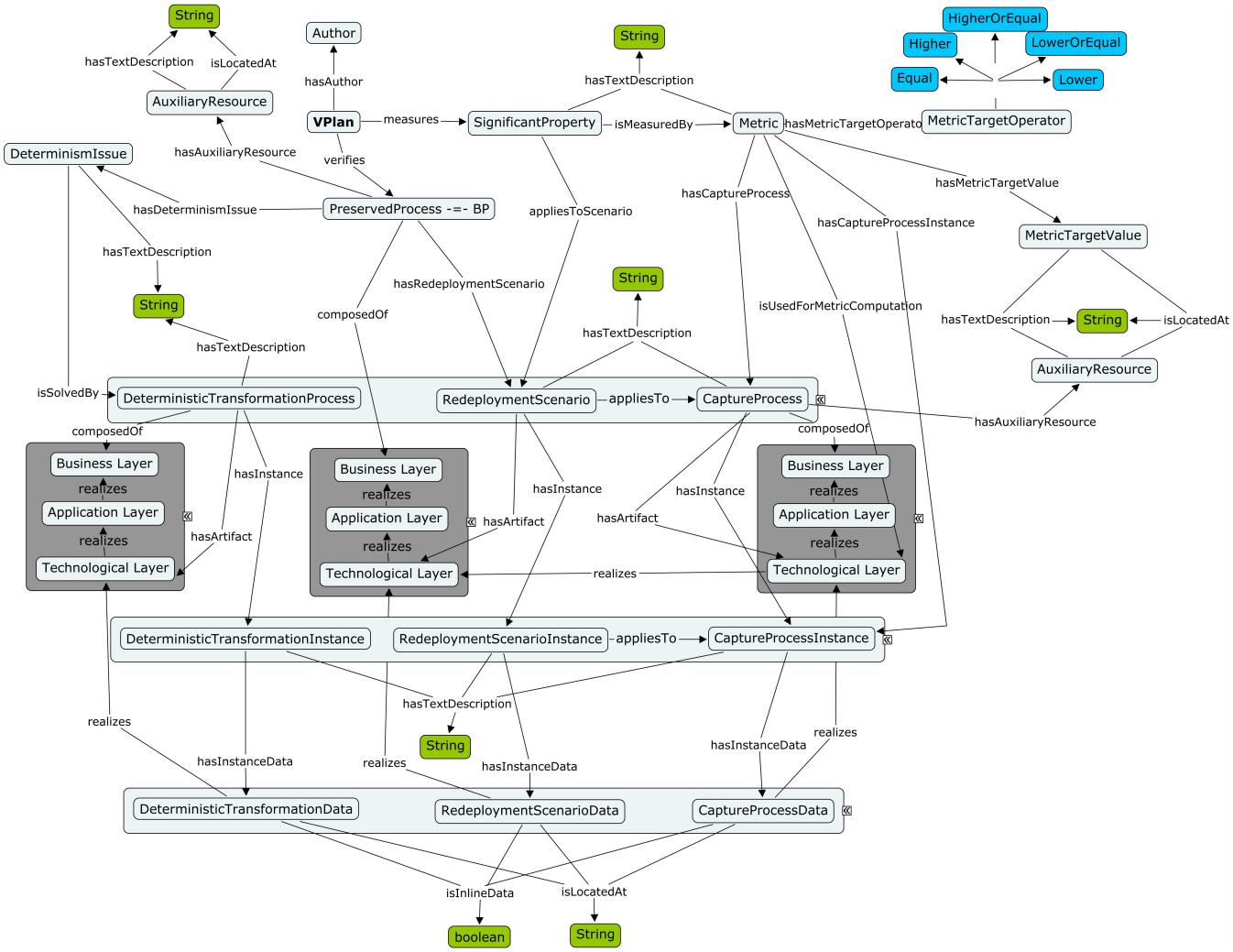

Figure 2: VPlan.

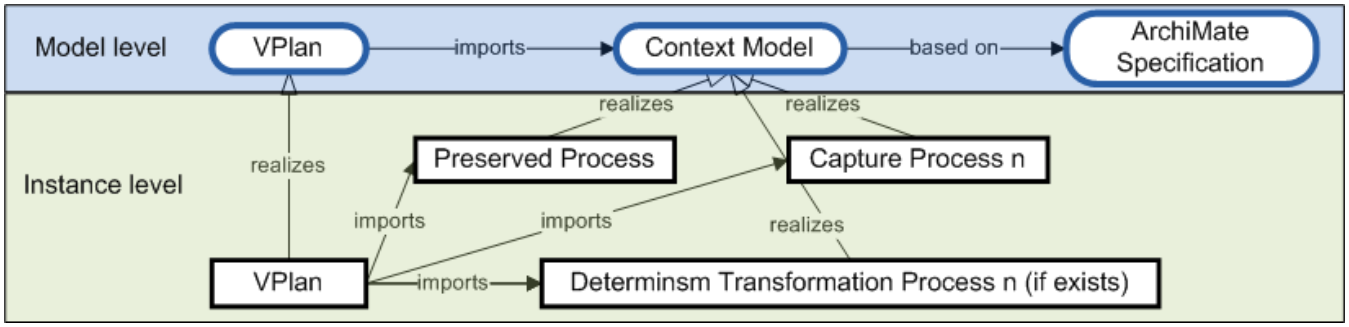

Figure 3: Differentiation between the VPlan model and the instance and an overview of imports made to the VPlan.

later used for metric computation, is extracted from the process. A similar approach was taken to the one from the Section 3.2.2 regarding the import of the preserved process model. Thus, each capture process is first modelled in ArchiMate, then converted to the ontology and finally imported to the VPlan.

Import of the capture process into the VPlan allows linking of the elements of the capture process with the elements of the preserved process. The link is essential, because in this way the generic process of capturing becomes concrete for the given preserved process. In other words, this link specifies the measurement point. For example, most of the capture processes provide at their output a file with some

data extracted from the process. In order to state from which part of the process and at which component the capturing took place, a link between the *CaptureProcess* and the *PreservedProcess* is established.

3.2.4 Import of determinism transformation processes at the instance level

When the process is not deterministic during its execution, i.e. has different characteristic and outputs for the same input data, then it is impossible to conduct faithful verification. The VFramework foresees such a situation and assumes that for the purpose of verification the process part which introduces the lack of determinism can be removed or substituted with a deterministic one. Due to this fact, the VPlan holds information on determinism transformation processes. These processes describe what has to be done in order to make the preserved process deterministic for the purpose of verification. Similar to the capture processes described in the section above, the determinism transformation processes are modelled in ArchiMate, using the Archi[7] tool, converted to ontology and then imported to the VPlan.

3.3 Mapping to the VFramework

In this section the mapping of the VFramework steps to the VPlan classes is presented. The aim of the mapping is to demonstrate, that the VPlan fulfils the requirements of the VFramework. For this reason, two figures depicting mapping of concepts in the original and in the redeployment environment were created and are discussed in the consecutive subsections.

3.3.1 Original environment

The VFramework steps that are executed in the original environment focus on collection of process information. At this phase the VPlan is created and filled with data. The Figure 4 depicts which VPlan classes are used at which step of the VFramework application. The numbers on the arrows depict the concrete steps and substeps of the VFramework. If all substeps of a given step of the VFramework are making use of a given class, then only a number of a step is provided on the arrow, e.g. *AuxiliaryResource* is used at all of the substeps of the "Describe the original environment" step of the VFramework, hence only 1 is used instead of 1.1/2/3/4.

In the first step of the VFramework, which is "Describe the original environment", not only the process and its context is described, but also the redeployment scenarios, verification instances and significant properties. According to the Figure 4 all these concepts are mapped to the respective classes.

In the second step of the VFramework, which is "Prepare system for preservation", a precise analysis of the process and its dependencies is conducted. This is the moment when the Context Model of the process is needed. The internal and external interactions of the process which are identified are modelled in the Context Model. The process boundaries are defined using *RedeploymentScenario* by specifying steps of the process that belong to the process. The deterministic behaviour is described using *DeterminismIssue* and a way of tackling it with a use of classes related to the transformation process.

[7]http://archi.cetis.ac.uk/

In the third step of the VFramework, which is "Design verification setting", the measurement points are specified by designing capture processes and linking them to the elements of the Context Model. The metrics for preservation quality comparison also have their respective classes for expressing the metrics and their value.

In the fourth step of the VFramework, which is "Capture verification data", the data is captured from the process by execution of process instances. The information on data location for each of the instances is also covered by the VPlan.

3.3.2 Redeployment environment

The VFramework steps, executed in the redeployment environment, focus on the actual verification of the redeployed process using the information collected in the original environment. At this phase the VPlan is accessed to read the information from it. The Figure 5 depicts which VPlan classes are used at which step of the VFramework. The convention used in the figure is similar to the one from the previous section. The only difference is the direction of the arrows which is opposite, since the information is read from the VPlan.

In the fifth step of the VFramework, which is "Prepare system for redeployment", the process is redeployed using information from the process Context Model. The process instances referred to by the VPlan are moved to the system in which they are executed.

In the sixth step of the VFramework, which is "Capture the redeployment performance data", the capture process which was used in the original environment is used to capture the information from the redeployed process. Sometimes repetition of the exact capture process is impossible, but it is up to the preservation expert to make a decision how to design a new capture process which is compatible with principles of the original one, which is provided by the VPlan.

In the seventh step of the VFramework, which is "Compare and asses", the final assessment of the redeployment is conducted. Information on metrics, their original values and expected values are obtained from the VPlan.

4. VPLAN EVALUATION

In this section we describe the application of the VFramework to an eScience use case. Section 4.1 details the use case. Section 4.2 explains how the VFramework was applied.

4.1 Use Case Description

The safety control of large dams is based on the monitoring of important physical quantities that characterize the structural behaviour (relative and absolute displacements, strains and stresses in the concrete, discharges through the foundations, and so on.). The analysis of data captured by the monitoring systems (sensor networks strategically located at dams) and their comparison with statistical, physical and mathematical models is critical for the safety control assessment. It is known that the variations of hydrostatic pressure and temperature are the main actions that must be considered when analysing the physical quantities generated by the monitoring systems. As a consequence, multiple linear

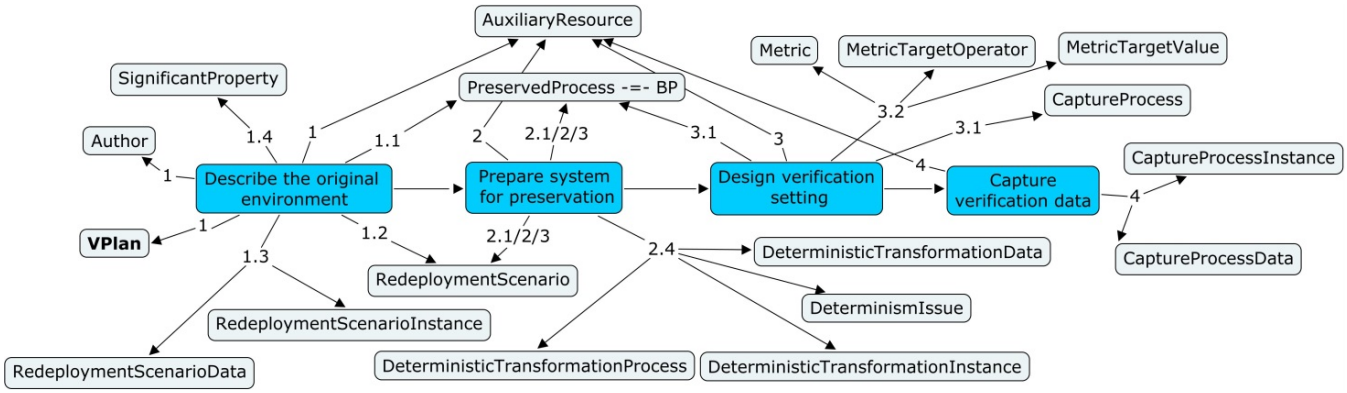

Figure 4: Mapping of the VPlan to the VFramework steps executed in the original environment.

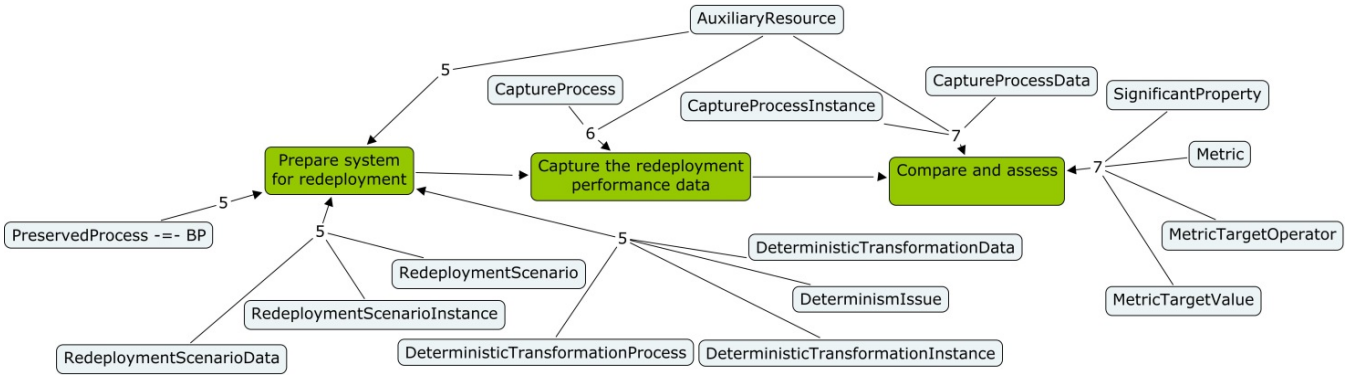

Figure 5: Mapping of the VPlan to the VFramework steps executed in the redeployment environment.

regressions (MLR) are highly suitable and efficient models to determine their relationship with the expected response (physical quantity)[10]. In fact, MLR models are used to model the linear relationship between a dependent variable (predictand or response) and one or more independent variables (predictors).

In large dams, the expected response is approximated by the following effects: (i) elastic effect of the hydrostatic pressures; (ii) elastic effect of temperature, depending on thermal conditions; and (iii) time effect (considered irreversible)[10]. The results of such models are used in structural safety to compare the estimated/predicted behaviour against the real behaviour (represented by the physical quantities captured from the monitoring systems)

Figure 6 details a multiple linear regression process used in dam safety to estimate the physical quantities based on the effects of hydrostatic pressure, temperature and time. For demonstration purposes, this process was isolated from the generic information system (*GestBarragens*). Overall, the process is composed of five steps:

- Extract data: Based on a set of extraction parameters, this process generates the sensor data that will be used in the MLR model (training set with historical values of independent and dependent variables).
- Generate regression: Based on a set of regression parameters (e.g., equation to estimate elastic effect of the hydrostatic pressure), this process generates the regression controls that configure the parameters for the MLR model.
- Execute regression: This process executes the regression parameterized in the regression control, using the training dataset generated in the extract data process. It generates a set of plots and tables to represent the results of the regression execution, including the coefficients (determine the linear relationship between the independent variables and the response, the quality measures (standard deviation, quadratic error, and so on.), residuals (fitting error), and the ANOVA matrix for variance analysis[8].
- Generate aggregation: since a dam has a large number of sensors and a regression is used for each physical quantity associated with each sensor, we might need to run hundreds or thousands of regressions. Thus, the process is able to aggregate all MLR executions into one aggregated report. This step generates the controls that define how this data is aggregated.
- Produce report: This collects all the results produced

[8]The coefficients are used to generate expected responses from the known independent variables. The quality measures, residuals and ANOVA matrix are crucial to determine if a specific MLR model is adequate to estimate and validate a specific physical quantity.

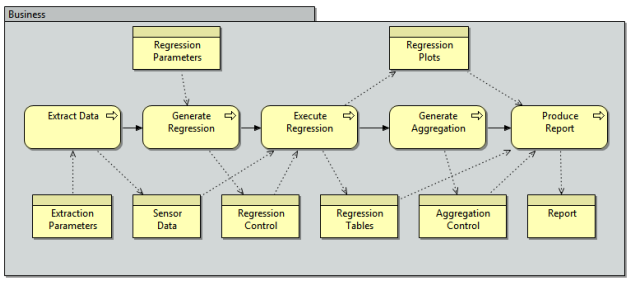

Figure 6: Multiple linear regression process in dam safety.

by the several executions of MLR models and compile them into a single report.

4.2 VFramework Application

As in Section 3.3, we first describe steps taken in the original environment (Section 4.2.1) and then in the redeployment environment (Section 4.2.2).

4.2.1 Original Environment

Following the VFramework, the initial steps have the purpose of collecting all data about the process we want to preserve. This involves initializing a clean ontology file to populate it with the process information. The ontology file will represent the VPlan. In the first step "Describe the original environment" we modelled the process that we want to preserve in ArchiMate using the Archi tool, and imported it to our VPlan. Figure 6 depicts the business layer of the process.

Before import, the process was detailed in terms of the application and technology layer. Note that the final model could also be enriched by the use of context extractors as, for instance, a hardware extractor to further detail the technology layer. It was also defined that the process is preserved with one redeployment scenario in mind. That scenario assumes that the process is fully redeployed to reproduce its original behaviour. One instance of the scenario was stored. Instance data simply consisted of the process application (represented by an executable file at the technological level) and extraction parameters (represented by an "app.config" file) since using the same parameters the application must always produce the same results.

In terms of significant properties that the process needs to maintain we identified and defined the following:

- SP1 - Generate data: the system must be able to generate sensor data for quantitative interpretation.
- SP2 - Export by: the system must generate data for a specific structure, date period and sensor type.
- SP3 - Quantitative interpretation: the system must be able to execute the quantitative interpretation for all the physical quantities of the selected sensor type.
- SP4 - Coefficients: the system must provide the coefficients used in the interpretation, mainly estimate, standard error, t value, Pr(>|t|).
- SP5 - Quality Measures: the system must provide the quality measures of the regression, mainly standard deviation, quadratic error and adjusted quadratic error.
- SP6 - Residuals: the system must provide the residuals of the regression in a table;
- SP7 - ANOVA Matrix: the system must provide the ANOVA matrix of the regression.
- SP8 - Report: the output of the process should be compiled into a single PDF report.

All this information was added to the VPlan. The state of the VPlan after execution of the first step is depicted in Figure 7.

In step two, "prepare system for preservation", the process was analysed in terms of dependencies and determinism. It was concluded that the process is indeed deterministic so there was no need to define a deterministic transformation process. The process has three dependencies on external web-services required to execute the process. We consider that the decision whether to preserve or not the web-services is out of the scope of the VFramework. Ideally stakeholders applying the VFramework should perform a risk analysis to understand whether the web-services are going to be available at redeployment or, if necessary, to preserve them along with the process. In this particular application we did not preserved the web-services and consequently no changes to the VPlan were necessary at this step.

Step three, "design verification setting" is all about assigning metrics to the significant properties and defining how those metrics should be captured. For each metric we defined a text description, a capture process, a target operator and, if applicable, a target value. The combination of the target operator and target value determines the required value of a metric to be considered successful. The absence of the target value indicates that the value of the metric at redeployment should be compared to the value at the original environment. Figure 8 illustrates the definition of a metric using the ontology-editor Protégé[9]. Figure 9 illustrates the capture process entitled "CaptureProcess6" that is defined on Figure 8. All capture processes were defined with the Archi tool, converted to the Context Model and added to the VPlan.

Metrics were associated with significant properties in the following way:

- For SP1, two metrics were defined. Both involve understanding whether "sensor data" generated by the "extract data" step of the process is the same at both the original and redeployment environment. To measure it, one of the metrics involves counting the number of files that were generated and the other consists of counting the number of lines in each file. For the

[9]http://protege.stanford.edu/

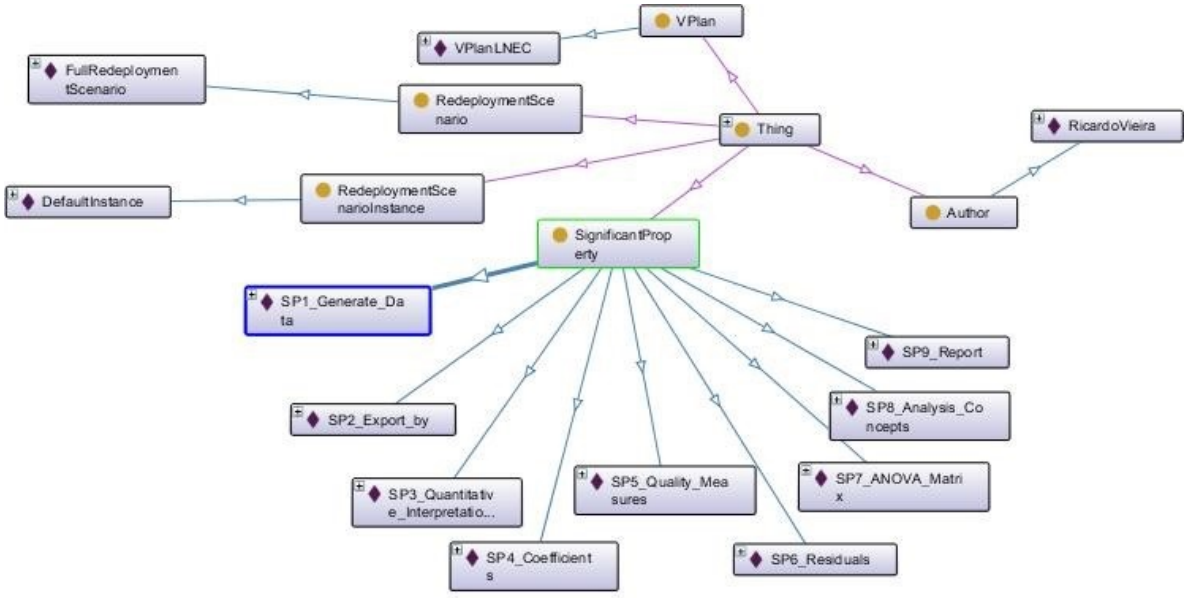

Figure 7: Simplified visualisation of the VPlan after the first step of the VFramework.

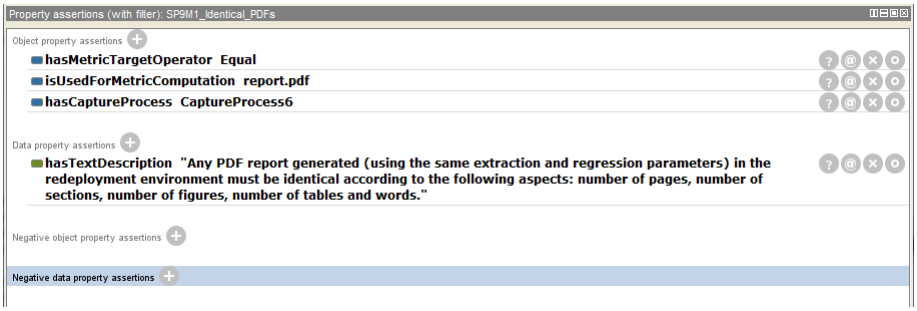

Figure 8: Example of a metric modelled in VPlan using Protégé.

same instance of the process, i.e. for each execution of the process using identical "extraction parameters", the numbers need to be equal in both environments.

- SP2 had three metrics. Both involve understanding if the generated data conforms to the "export by" filter. To measure it, we check if the generated data contains data that must not be exported, namely: (1) data from a dam that was not specified; (2) data from a data outside of the selected data period; or (3) data from a sensor not belonging to the selected sensor type.
- SP3 and SP8 had similar metrics. Both properties had one metric and required the execution of the step "execute regression". That specific step generates "regression plots". SP3 metric involves checking if a plot is generated for each physical quantity present in the sensor data. SP8 metric involves checking if a graphical representation is generated for each analysis concepts (10 concepts in total).
- SP4 to SP7 also have one metric each defined. Again, the capture process involves the execution of the step "execute regression" but now requires the verification of the generated "regression tables". The metrics will verify, respectively, if the "regression tables" have all coefficients, quality measures, residuals, and ANOVA Matrixes.
- SP9 has one metric to verify if the report generated at the end of the process is equal both in original and redeployment environment. As illustrated in Figure 8 the metric compares the report in terms of number of pages, sections, figures, tables and words.

In the last step at the original environment "capture verification data" we executed the previous defined capture process and stored the required files. Note that only SP1 and SP9 require comparison between original and redeployment environment so only those capture process were performed at the original enviroment.

4.2.2 *Redeployment Environment*

The fifth step of the VFramework which is "prepare system for redeployment" involves redeploying the process using the information stored in the VPlan. As in [13], since the preserved process depends on Microsoft .NET Framework 4.0, for redeployment we opted to use a machine running

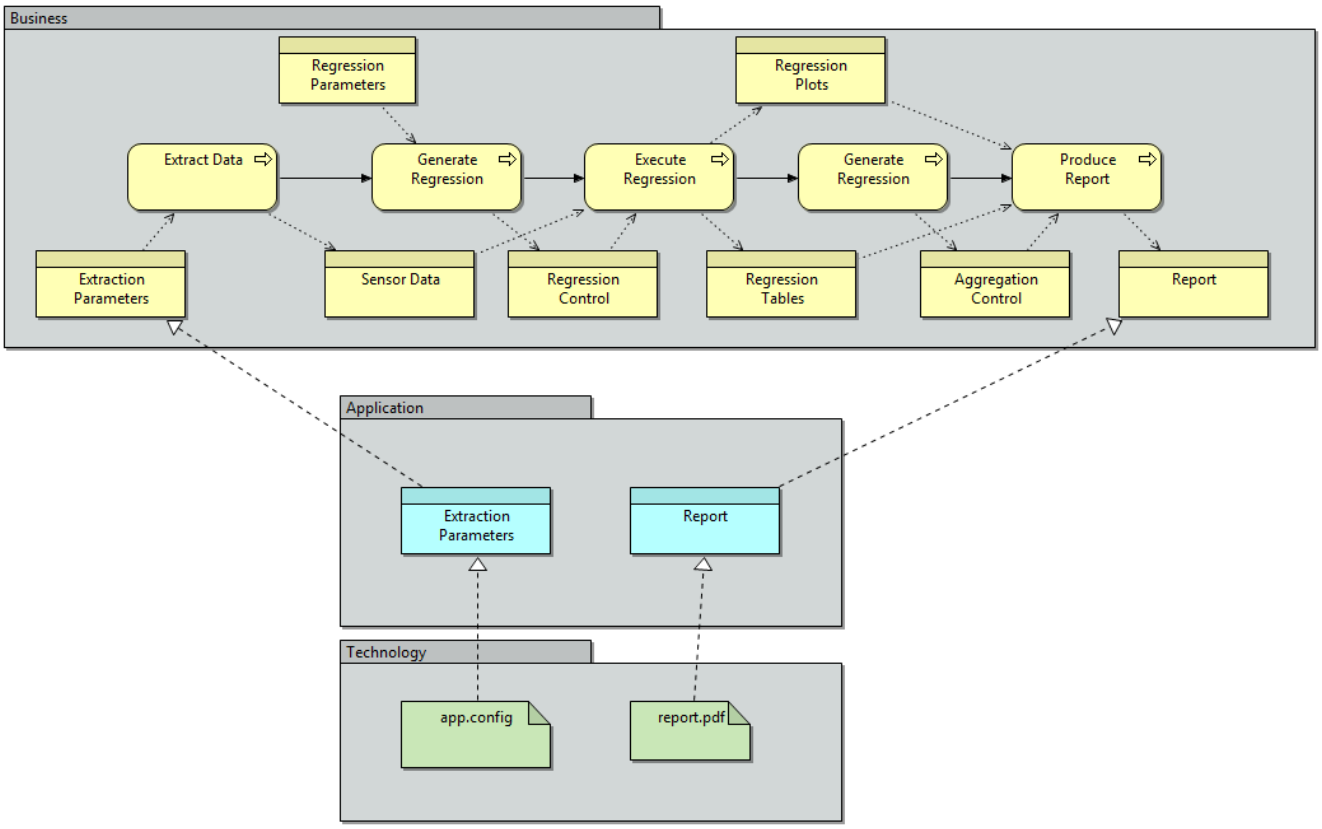

Figure 9: Example of a capture process modelled in Archi.

Ubuntu Linux[10] 12.10 - an open source operating system based on the GNU Linux kernel, which allows us to simulate a slightly different redeployment environment. However, since the .NET platform is exclusively available for Microsoft operating systems, several challenges had to be addressed to re-execute the process in Lunux (for more information refer to [13]).

In the sixth step "Capture redeployment performance data", the capture processes defined in the third step "design verification setting" were executed in the redeployment environment. All processes were executed manually. The result of the execution was a set of files, each associated to a specific metric, that are required for verification of the metric. As an example, the last metric (from SP9) involved executing all the steps of the process and storing the final report for metric assessment in the next step.

Finally, in the last step "compare and assess" we compared all the results of the capture process to assess if the significant properties were maintained. We consider process to have retained a specific significant property when all of the metrics associated with it are successful verified. To assess a metric we require the target operator and target value (if one exists) from the VPlan in order to understand the type of comparison that needs to be performed and the expected value. All metrics were successfully verified so we concluded that all significant properties from the original environment were maintained at redeployment. Continuing our example, in the metric from SP9 the target operator is "equal" and there is no target value (as illustrated in Figure 8) meaning that it is necessary to compare data from the original environment (captured in step 5 - "capture verification data") with data from the redeployment environment (captured in the previous step). In this specific example we needed to compare two reports, represented as PDF files, in terms of number of pages, sections, figures, tables, and words. Both reports had 25 pages, 5 sections, 80 figures, 33 tables and 1660 words allowing the conclusion that the metric is valid and SP9 was maintained.

[10]http://www.ubuntu.com/

5. CONCLUSIONS AND FUTURE WORK

In this paper the VPlan ontology for collection of process verification data was presented. It allows storing information on significant properties, metrics, capture processes and data collected during the verification of preserved and redeployed processes with a use of the VFramework. The VPlan increases the confidence that the evidence needed for the verification of processes is properly organized and stored.

When introducing the VPlan we described its structure (classes and properties) and its integration with the TIMBUS Context Model. Moreover, we provided a mapping of the VPlan concepts to the VFramework in order to demonstrate that the VPlan addresses all of the requirements of the VFramework. Finally, we showed how the VPlan facilitates the ver-

ification of preserved and redeployed process by applying it to a typical data analysis process from a civil engineering domain.

We are currently working on automation of VPlan creation, so that some of its parts can be automatically generated. This should increase the acceptance within the scientific community. We are also developing a set of SPARQL queries which not only validate the VPlan, but also facilitate retrieval of the information stored in the VPlan. Future work will also focus on further testing on different use cases.

ACKNOWLEDGMENTS

This research was co-funded by COMET K1, FFG - Austrian Research Promotion Agency and by the European Commission under the IST Programme of the 7th FP for RTD - Project ICT 269940/TIMBUS.

6. REFERENCES

[1] R. S. Aguilar-Savén. Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2):129 – 149, 2004. Production Planning and Control.

[2] G. Antunes, M. Bakhshandeh, R. Mayer, J. Borbinha, and A. Caetano. Using ontologies for enterprise architecture analysis. In *Proceedings of the 8th Trends in Enterprise Architecture Research Workshop (TEAR 2013), in conjunction with the 17th IEEE International EDOC Conference (EDOC 2013)*, Vancouver, British Columbia, Canada, September 9-13 2013.

[3] K. Belhajjame, O. Corcho, D. Garijo, J. Zhao, P. Missier, D. Newman, R. Palma, S. Bechhofer, E. Garcia Cuesta, J. M. Gomez-Perez, S. Soiland-Reyes, L. Verdes-Montenegro, D. De Roure, and C. Goble. Workflow-centric research objects: First class citizens in scholarly discourse. In *Proceedings of Workshop on the Semantic Publishing*, 2012.

[4] C. Collberg, T. Proebsting, G. Moraila, A. Shankaran, Z. Shi, and A. Warren. Measuring Reproducibility in Computer Systems Research. Technical report, 2013.

[5] M. Guttenbrunner and A. Rauber. Evaluating an emulation environment: Automation and significant key characteristics. In *Proceedings of the 9th International Conference on Digital Preservation (iPres 2012)*, pages 201–208, Toronto, Canada, October 1-5 2012.

[6] M. Guttenbrunner and A. Rauber. A measurement framework for evaluating emulators for digital preservation. *ACM Transactions on Information Systems (TOIS)*, 30(2), 3 2012.

[7] V. Haren and V. H. Publishing. *ArchiMate 2. 0 Specification*. The Open Group. Van Haren Publishing, 2012.

[8] K. Hettne, S. Soiland-Reyes, G. Klyne, K. Belhajjame, M. Gamble, S. Bechhofer, M. Roos, and O. Corcho. Workflow forever: Semantic web semantic models and tools for preserving and digitally publishing computational experiments. In *Proceedings of the 4th International Workshop on Semantic Web Applications and Tools for the Life Sciences*, SWAT4LS '11, pages 36–37, New York, NY, USA, 2012. ACM.

[9] S. Jones. A report on the range of policies required for and related to digital curation. Technical Report 1, Mar. 2009.

[10] J. Mata. Interpretation of concrete dam behaviour with artificial neural networks and multiple linear regression models. *Engineering Structures*, 33(3):903–911, 2011.

[11] R. Mayer, A. Rauber, M. A. Neumann, J. Thomson, and G. Antunes. Preserving scientific processes from design to publication. In P. Zaphiris, G. Buchanan, E. Rasmussen, and F. Loizides, editors, *Proceedings of the 16th International Conference on Theory and Practice of Digital Libraries (TPDL 2012)*, volume 7489 of *Lecture Notes in Computer Science*, pages 113–124, Cyprus, September 23–29 2012. Springer.

[12] B. D. Mccullough. Got Replicability? The Journal of Money, Credit, and Banking Archive. *Econ Journal Watch*, 4(3):326–337, Sept. 2007.

[13] T. Miksa, S. Proell, R. Mayer, S. Strodl, R. Vieira, J. Barateiro, and A. Rauber. Framework for verification of preserved and redeployed processes. In *Proceedings of the 10th International Conference on Preservation of Digital Objects (IPRES2013)*, Lisbon, Portugal, September 2–6 2013.

[14] T. Miksa and A. Rauber. Increasing preservability of research by process management plans. In *Proceedings of the 1st International Workshop on Digital Preservation of Research Methods and Artefacts*, DPRMA '13, pages 20–20, New York, NY, USA, 2013. ACM.

[15] P. Missier, S. S. Sahoo, J. Zhao, C. A. Goble, and A. P. Sheth. Janus: From Workflows to Semantic Provenance and Linked Open Data. In *Proceedings of the International Provenance and Annotation Workshop (IPAW2010)*, pages 129–141, Troy, New York, USA, June 15–16 2010.

[16] O. M. G. (OMG). Business process model and notation (bpmn) version 2.0. Technical report, jan 2011.

[17] S. Strodl, D. Draws, G. Antunes, and A. Rauber. Business process preservation, how to capture, document & evaluate. In *Proceedings of the 9th International Conference on Preservation of Digital Objects (IPRES2012)*, Toronto, Canada, October 2012.

[18] S. Strodl, R. Mayer, D. Draws, A. Rauber, and G. Antunes. Digital preservation of a process and its application to e-science experiments. In *Proceedings of the 10th International Conference on Preservation of Digital Objects (IPRES 2013)*, 9 2013.

[19] J. S. Susanne Glissman. A comparative review of business architecture. Technical report, IBM Research Division, August 24 2009.

[20] The Open Group. *Archimate 2.0: A Pocket Guide*. TOGAF series. Van Haren Publishing, 2012.

[21] TIMBUS Consortium. D4.6: Use Case Specific DP & Holistic Escrow. Technical report, 2013.