Repeatability and Re-usability in Scientific Processes: Process Context, Data Identification and Verification

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

Tomasz Miksa, Rudolf Mayer, Stefan Proell
SBA Research
Vienna, Austria
Email: {tmiksa, rmayer, sproell}@sba-research.org

***Abstract*—eScience offers huge potential of speeding up scientific discovery, being able to flexibly re-use, combine and build on top of results. Yet, in order to reap the benefits promised by eScience, we must be able to actually perform these activities, i.e. having the data, processing components available for re-deployment. Furthermore, repeatability of e-Science experiments is widely understood as a requirement of validating work to establish trust in results obtained, specifically in data-intensive domains. This proves challenging as procedures currently in place are not set up to meet these goals. This renders repeatability a challenging task.**

A number of approaches have tackled this issue from various angles. This paper reviews several of these building blocks and ties them together. It starts from the principles of data management plans. We review their strengths and weaknesses and outline ways to address them. We then move beyond data, addressing the capture and description of entire research processes, ways to document and prepare them for archival. We review the recommendations of the Research Data Alliance on how to precisely identify arbitrary subsets of potentially high-volume and highly dynamic data used in a process. Last, but not least, we present mechanisms for verifying the correctness of process re-executions.

I. Introduction

The advent of new means of performing research and sharing results offers huge potential for speeding up scientific discovery, enabling scientists to flexibly re-use, combine and build on top of results without geographical or time limitations and across discipline boundaries. Yet, in order to reap the benefits promised by eScience [1], we must be able to actually perform these activities, i.e. having the data, processing components available for re-deployment. The G7[1] as well as funding agencies such as the EC[2] are commited to data re-use and open data initiatives. As a result, all research data from publicly funded projects needs to be made available for the public. Not only does this entail that the data must be eqipped with useful and stable metadata, comprehensive descriptions and documentation, but also that the data must be preserved for the long term.

From a scientific point of view, the validation of research results is a core requirement, which is needed for establishing trust in the scientific community, specifically in data-intensive domains. This proves challenging as procedures currently in place are not set up to meet these goals. Experiments are often complex chains of processing, involving a number of data sources, computing infrastructure, software tools, or external and third-party services, all of which are subject to change dynamically. In scientific research external influences can have a large impact on the outcome of an experiment. Human factors, the used tools and equipment, the settings and configuration of the used soft- and hardware, the execution environment and its properties are important factors which need to be considered. The impact of such dependencies has proven to be more grave than might be expected. While many approaches rely on documenting the individual processing steps performed during an experiment, on storing the data as well as the code used to perform an analysis, the impact pof the underlying software and hardware stack are often ignored. Yet, beyond the challenges posed by the actual experiment/analysis, it is the complexity of the computing infrastructure (both the processing workflows and their dependencies on HW and SW environments, as well as the enormous amounts of data being processed) that renders research results in many domains hard to verify. As a recent study in the medical domain has shown [2], even assumedly minute differences such as the version of the operating system used can have a massive impact: different results were obtained in cortical thickness and volume measurements of neuroanatomical structures if the software setup of FreeSurfer, a popular software package processing MRI scans, is varied. More dramatically, though, there is also a difference in the result if not the primary software, but only the operating system version (in this case the Mac-OSX 10.5 and 10.6) differ. This indicates the presence of dependencies from FreeSurfer to functions provided by the operating system, causing instabilities and misleading results. As these dependencies are hidden from the physician, such side-effects of the ICT infrastructure need to be detected and resolved transparently if we want to be able to trust results based on computational analyses.

Two fundamental concepts of research are repeatability and reproducibility, which describe the circumstances under which an experiment must deliver the same results in order to be verifiable. An experiment is repeatable, if it produces the exact same results under the very same preconditions. An experiment is reproducible, if the same results can be obtained even under

[1] https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/207772/Open_Data_Charter.pdf

[2] http://ec.europa.eu/digital-agenda/en/open-data-0

somewhat different conditions, e.g. performed by a different team in a different location.

A number of approaches have tackled this issue from various angles, including initiatives for data sharing, code versioning and publishing as open source, the use of workflow engines to formalise the steps taken in an experiment, to ways to describe the complex environment an experiment is executed in. In addition the data that is created but also the processig alogorithms, scripts, and other software tools used in the experiment need to be accessible for longer time periods, for facilitating data reuse and allowing peers to retrieve and verify experiments. Keeping these assets accessible is not only a technical challenge, but requires institutional commitment and defined procedures.

Repeatability and reproducibility are two fundamental concepts in science. Both principles allow peers to verify the correctness of results by executing an experiment again. There are several factors which have an influence on the variance of experiments. The ISO standard 5725-1:1994 [3] lists the following factors: (1) operator, (2) equipment, (3) calibration of the equipment, (4) environment and (5) time elapsed between measurements. The standard defines an experiment as repeatable, if the mentioned influences (1) - (4) are constant and (5) is a reasonable time span between two executions of the experiment and its verification. Reproducibility in contrast allows variance in the factors, as they cannot be avoided if different research teams want to compare and verify results.

In order to tackle these issues we proposed to introduce Process Management Plans (PMPs) in [4]. This solution extends existing Data Managament Plans by taking a process centric view, viewing data simply as the result of underlying processes such as capture, (pre-) processing, transformation, integration and analyses. The general objective of PMPs is to foster identification, description, sharing and preservation of scientific processes. In order to embody the concept of PMPs we need to solve the challenges related to the description of computational processes, verification and validation, monitoring external dependencies, as well as data citation have to be solved.

This paper reviews these building blocks and ties them together in order demonstrate that sharing and preservation of not only datasets, but also scientific processes is possible.

Section II provides an overview of existing Data Managament Plans and describes existing tools for data and process sharing. In Section III we specify what information is included in a PMP. Section IV focuses on a Context Model used that is automatically captured and used for description of the process implementation in one of the parts of the PMP. Data Citation is also one of the building blocks of the PMP that is descibed in Section V. Section VI presents methodology and challenges related to verification of computational processes. These concepts are illustrated via a use case from the machine learning domain in Section VII, followed by conclusions in Section VIII.

II. Related Work

This section presents related work in the domains of data management, digital preservation, eScience and research infrastructures.

A. Data Management Plans

A prominent reason for the non-reproducibility of scientific experiments is porr data management, as criticized in several disciplines. Different data sets scattered around different machines with no track of dependency between them are a common landscape for Particle Physicists who move quickly from one research activity to another [5]. Several institutions reacted, publishing templates and recommendations for DMPs, such as the Digital Curation Centre (DCC) [6], Australian National Data Services (ANDS) [7] and National Science Foundation (NSF) [8], amongst many others. These are very similar, containing a set of advices, mainly lists of questions which researches should consider when developing a DMP. The attention is attracted to what happens with data after it has been created, rather than in what way it was obtained. All the description is provided in a text form, and in case of NSF there is a limit of 2 pages. Therefore the possibility to reuse or at least reproduce the process which created the data is very unlikely. Furthermore, the correctness of data is taken for granted and thus DMPs do not provide sufficient information that would allow validating the data. Finally, the quality and detail of information strongly depends on the good will of researchers. There is no formal template for specification of DMPs which would ensure that all important information is covered comprehensively. Several tools are available, like *DMPonline*[3] for DCC or *DMPtool*[4] for NSF, which aid the researcher in the process of DMP creation, but they are rather simple interactive questionnaires which generate a textual document at the end, rather than the complex tools required to validate at least the appropriateness of the provided information. The main conclusion from the analysis is that DMPs focus on describing results of experiments. This is a consequence of their data centric view, which enforces focus on access and correct interpretation (metadata) of data and does not pay much attention to processing of data. While these constitute an extremely valuable step in the right direction, we need to move beyond these initial steps, taking a process centric view. This complements the advantages of DMPs (documenting data) by adding information on the processes which created the data.

B. Digital Preservation

The area of digital preservation is shifting focus from collections of simple objects to the long term preservation of entire processes and workflows. There are a number of research projects addressing the challenges of keeping processes available in the long term. Tools, methods and other research outputs, which may be used to ensure processes can be maintained accessible and useable.

WF4Ever[5] addressed the challenges of preserving scientific experiments by using abstract workflows that are reusable in different execution environments [9]. The abstract workflow specifies conceptual and technology-independent representations of the scientific process. They further developed new approaches to share workflows by using an RDF repository and make the workflows and data sets accessible from a SPARQL

[3]https://dmponline.dcc.ac.uk/

[4]https://dmp.cdlib.org/

[5]http://www.wf4ever-project.org/

Endpoint[10]. The TIMBUS[6] project addressed the preservation of business processes by ensuring continued access to services and software necessary to properly render, validate and transform information. The approach centers on a context model [11] of the process, which is an ontology for describing the process components and their dependencies. It allows to store rich information, rainging from software and hardwareto organisational and legal aspects. The model can be used to develop preservation strategies and redeploy the process in a new environment in the future. The project developed a verification and validation method for redeployed processes [12] that evaluates the conformance and performance quality processes redeployed in new environments. This is especially important when PMPs are used for the purpose of validation (by re-executing the process), or reuse (to build other process).

C. eScience and Research Infrastructures

Several projects benefit nowadays from sharing and reusing data [13]. An example of successful sharing of data is the Economic and Social Data Service [14] provided by Economic and Social Research Council in Great Britain. The study proved that the value of shared data to the researchers is "$25 million per annum at 2010 prices and levels of activity use". This confirms that properly managed and shared data can result in major benefits. In [15] the evolution of research practices by sharing of tools, techniques and resources is discussed. *myExperiment* [16] is a platform for sharing of scientific work-flows. This is already one step beyond just sharing the data. Workflows created and run within the *Taverna* workflow engine can be published and reused by other researchers. However, the workflows do not always specify all required information (e.g. tools to run the steps, description of parameters) to re-run the workflow [17]. Process Management Plans address this problem, ensuring that all necessary information is provided. They also put a much stronger focus on the research process than found within so-called executable papers.

An environment which enables scientists to collaboratively conduct their research and publish it in form of executable paper was presented in [18]. The solution requires working in a specific environment, limiting its applicability to the tools and software supported by the environment. PMPs does not have such a requirement and can be used in every case. The necessity to introduce the PMPs is also driven by the rising number of scientific experiments using specialised middleware and infrastructure. One of the efforts aiming to provide such an infrastructure is described in [19]. The authors describe steps towards "providing a consistent platform, software and infrastructure, for all users in the European Research Area to gain access to suitable and integrated computing resources".

III. Process Management Plan

This section presents a proposed structure of Process Management Plans (PMPs) and is based on [4]. A PMP is a living document that is more than just a paper report (or a digital version of it). In fact, in order to make the PMPs actionable and enforceable, automation of its creation and machine readability are required. This helps to ensure higher precision and coherence of information included in a PMP. Moreover, it decreases the time required to create it if the information is collected automatically. Tools, methods and concepts which facilitate implementation of PMPs are presented in Sections IV, V and VI.

The proposed structure of the PMP, following and extending guideliness for Data Management Plans, is presented below.

1) Overview and context
2) Description of the process and its implementation
   - Process description
   - Process implementation
   - Data used and produced by process
3) Preservation
   - Preservation history
   - Long term storage and funding
4) Sharing and reuse
   - Sharing
   - Reuse
   - Verification
   - Legal aspects
5) Monitoring and external dependencies
6) Adherence and Review

A. Overview and context

This section of PMPs provides a high level overview of the research activity and its context. It allows quick identification of what the project is about, who is involved in it and what the requirements and constraints set to the research project are. This information should follow a precisely defined schema for automated analysis and processing. It should cover things like project name, funding body, budget, duration, research objectives, list of requirements and policies which influence the creation of the PMP, list of involved people and organisations, state of the PMP, etc.

B. Description of the process and its implementation

Each process used during the course of research must be described. The description consists of three main parts presented below.

1) Process description: Process descriptiona are provided at different levels of details. An executive summary allows quick understanding of the purpose of the process, anhanced by more detailed descriptions of steps, data used, research methods. These should follow best practices used within the given scientific community. It should include specification of both functional and non-functional characteristics of a process, as well as any auxiliary resources which help to understand the process, e.g. publications, slides, tutorials, etc.

2) Process implementation: In order to analyse and reuse the process its structure must be understood and documented. This implies that all components that are used within a process implementation, their dependencies and relations between them have to be discovered and documented. The infrastructure used to run the experiment and specific software and hardware needed to run the process, e.g. special database software, libraries, software device drivers, fonts, codecs, have to be covered in this PMP section. It is essential to capture the full

[6] http://timbusproject.net/

process context including all the dependencies and relations between them, as this information is crucial for reusing the process, as well as ensuring its continuity by applying digital preservation actions.

3) Data used and produced by process: References to data used in the process have to be provided. This part links to an accompanying Data Management Plan providing information on input, intermediate and result data of the process. For example, information on the data formats used in the process may help researchers to decide if they can easily reuse the process with their own datasets. Existing templates for DMP specification can be reused to provide this information. Furthermore, techniques providing unambiguous identification of data sets and allowing data citation are needed.

C. Preservation

Planning for long term storage and securing funding for this purpose in advance increases the confidence that the research results will be available in the future. Two kinds of information concerning preservation and required by PMPs are discussed below.

1) Preservation history: PMP is a living document edited by multiple stakeholders. Therefore it collects information on actions that are performed to maintain the process over time. This information can be provided by the repository which takes care of a long term availability of the process. For example, outdated or obsolete hardware may be emulated; data may be migrated to a new format and the part of the process in which the process reads the data is newly implemented or substituted with a similar software [20]. A full track of changes to the original process implementation and evidence that these actions were performed correctly is necessary to maintain the authenticity of the process. Such information can be automatically obtained from tools which assist preservation planning [21].

2) Long term storage and funding: The sustainability of research results is increased by depositing the process in a trusted repository. The information on how long the research object will be kept is specified in this section of the PMP. Some parts of process may be discarded after certain time periods, keeping only some artefacts. People or institutions responsible after the end of a research project for taking decisions about the deposited process have to be assigned. Finally, information on funding of actions ensuring sustainability of processes (e.g. preservation actions or costs of storage) is specified using one of the available cost models [22].

D. Sharing and reuse

To support sharing and the reuse of results PMPs need to provide verification methods and data which can be used to verify if a process behaves like the original process upon redeployment. Finally, PMPs provide information on legal regulations and ethical issues related to the process.

1) Sharing: The process, its implementation and documentation about it are stored and means of providing access are specified. Conditions on which the resource can be accessed is provided, e.g. if the access is free or paid. Besides this, information on where the research results are published and how the location of the process is disseminated (e.g. scientific paper, blog, presentations, etc.) has to be given. If the process cannot be shared (e.g. due to non-disclosure agreements), then this information has to be provided here.

2) Reuse: There are many possible process reuse scenarios, for example: rerunning the original experiment, applying a process to new data, reproducing the experiment with improved computation algorithms or tools, reusing part of the experiment to build a new experiment, etc. In every case, the process must be ported, installed and configured in a specific environment. Although the comprehensive description provided in previous sections provides exhaustive information on the process and its dependencies, it may still not be sufficient for setting up and reusing the process. Therefore, a list of actions which help to port, install and configure the process on a new platform is needed.

3) Verification: Before using a process that is run again after being preserved we need to verify its conformance to the original behaviour. Therefore a set of precisely described tests showing process conformance is described in this section.

4) Legal aspects: This section focuses on legal aspects of working with the process, collecting information on licenses, copyrights of data and software. Legal regulations affecting the reuse of a process orethical or privacy issues (e.g. confidentiality of data) which may restrict use or distribution of entire processes or its parts are described here.

E. Monitoring and external dependencies

Processes are implemented in a specific environment which must be available in order to run the processes. PMPs specify the process components needed to run the process and aim to ensure that information about them is available. For example, if the process uses external web service to import some data, then this web service has to be monitored for its availability. Otherwise, if the web service is no longer available, the process is not operable. Therefore, PMPs should specify a list of critical dependencies which should be periodically monitored for their availability. The monitoring can be performed relying on concepts of Resilient Web Services [23].

F. Adherence and Review

Due to the fact that the PMP is a living document, it must be kept up to date and must reflect the actions that took place in each of its lifecycle phases. In order to ensure adherence, a person responsible has to be assigned. Furthermore, the reviews and methods applied to ensure that the PMP reflects the reality have to be in place. This information needs to be specified at a very early stage of PMP development and is highly dependent on the implementation of the PMP. As already mentioned, machine actionability of the PMP can foster its enforceability. This can be achieved when the PMP is not a static textual report, but a structured documented partially generated and interpreted by a machine.Otherwise, manual reviews conducted by auditors are needed and have to be planned in advance. The outcome of inspections has to be included in the PMP as well.

IV. Description of Processes

To enable analysis, repeatability and reuse of processes, they must be well described and documented. As most processes are rather complex in their nature, a precise description is needed in order to re-enact the execution of the process. Thus, formalised models are useful for a detailed representation of critical aspects such as the hardware, software, data and execution steps supporting the process, as well as their relationships and dependencies to each other.

Several different models can be considered for this type of documentation. Enterprise architecture (EA) modelling languages provide a holistic framework to describe several aspects of a process. For example, the Archimate [24] language supports description, analysis and visualisation of the process architecture, on three distinct but interrelated layers: business, application and technology layer. On each of these layers, active structures, behaviour and passive structures can be modelled. Thus the process can be specified not only as a high level sequence of steps, but also as a low level sequence of inputs and outputs from software and hardware components needed to run the process, e.g. special database software, libraries, software device drivers, fonts, codecs, or dedicated hardware created for the purpose of the experiment. Enterprise architectures do not address any specific domain-dependent concerns. They rather cut across the whole organisation running the process [25]. It is thus a major driver when designing a holistic model of a process, including the social, legal, organisational and technical environment it is embedded in.

Workflow-Centric Research Objects [26] (ROs) are a means to aggregate or bundle resources used in a scientific investigation, such as a workflow, provenance from results of its execution, and other digital resources such as publications, data-sets. In addition, annotations are used to further describe these digital objects. The model of Research Objects is in the form of an OWL ontology, and incorporates several existing ontologies. At its core, the Research Object model extends the *Object Exchange and Reuse* model (ORE) [27][7] to formalise the aggregation of digital resources. Annotations are realised by using the Annotation Ontology (AO) [28], which allows e.g. for comment and tag-style textual annotations. Specifying the structure of an abstract workflow is enabled by the *wfdesc* ontology. Finally, the provenance of a specific execution of a workflow is described using the *wfprov* ontology. Research objects have also been presented as a means to preserve scientific processes [29], proposing archiving and autonomous curation solutions that would monitor the decay of workflows.

While models such as Archimate or Research Objects are extensive, they often do not provide enough detail on technology aspects of the process, and thus in these aspects provide only little guidance to researchers aiming to produce a solid description of their technical infrastructure. One approach to alleviate this issue is realized in the Process Context Model [30], which builds on top of Archimate and extends it with domain specific languages to address specific requirements of a given domain. Wherever possible, the extension ontologies are based on already existing languages. The development of the model was driven by requirements to preserve and re-execute complete processes. The context the process is embedded in is assumed to range from immediate and local aspects such as the software and hardware supporting the process, to aspects such as the organisation the process is executed in, the people involved, service providers, and even laws and regulations. The exact context can differ significantly depending on the domain the process stems from.

[7] http://www.openarchives.org/ore/1.0

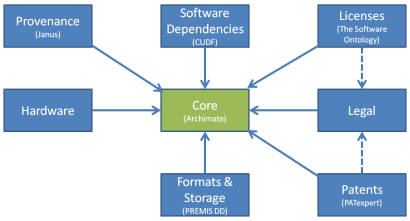


Fig. 2. Overview on the Context Model architecture: extensions and their relation to the core ontology

The model is thus using the domain-independent Archimate language as a core model to integrate the domain specific extension languages. It is currently implemented in the Web Ontology Language (OWL) [31], and the integration is performed via ontology mapping, from the extensions to the core model. An overview of this architecture and the provided domain-specific extensions is given in Figure 2. These are described in detail below.

Software Dependencies cover dependencies between different types of software, including information on which versions are compatible or conflicting with each other. It is, for example, important to know that a specific version of a Java Virtual Machine is required to run a certain piece of software, or that a particular application is required to view a digital object. This is important when considering preservation of specific parts of the software stack utilised in the process. Beyond repeatability, this information may be used during preservation planning to identify alternative software applications that can be utilised. Technical dependencies on software and operating systems in the Context Model can be captured and described via the Common Upgradeability Description Format (CUDF) [32].

Data Formats In a process execution, a number of digital objects are created, modified or read. This section includes information on which data/file formats these are stored in. Information on the format of these objects is crucial for digital preservation actions to be carried out, as e.g. migration to a different format might require changes in the rest of the process. The Context Model uses the PREMIS Data Dictionary [33] to represent this information.

Hardware contains a comprehensive description of the computational hardware, from desktop systems, server infrastructure components, to specialised hardware used for certain tasks. Even though in many processes the hardware employed to host the software applications might be standard commodity hardware, its exact specifications can still influence the runtime behaviour of a process. This might be critical in certain circumstances, such as execution speed, or when specific functionalities and characteristics of the hardware such as precision limits, analog/digital conversion thresholds etc. are part of the computation. Further, certain processes might utilise certain hardware capabilities for computation, such as using

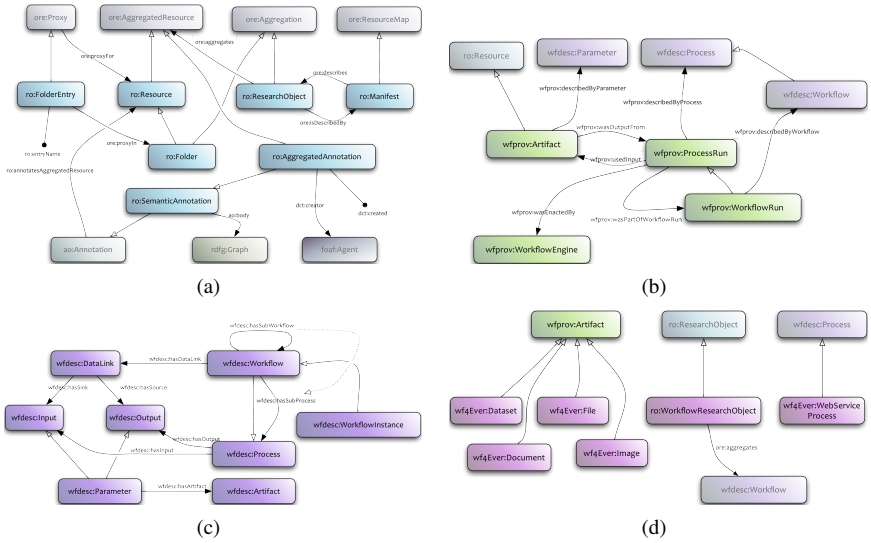


Fig. 1. Research Object Ontologies Core model, Workflow description model, Workflow provenance model, Workflow4Ever specific model

graphical processing units (GPUs) for large-scale experiments in scientific processes. These types of hardware, and the software that can work upon them, are not yet as standardised and abstracted, thus an exact description is needed in many cases.

Legal aspects includes all legal requirements imposed on the processes and surrounding context. While the domain is very extensive, we consider the following fields of specific importance.
License information includes all aspects related to licenses, focusing specifically on software licenses. Relevant aspects are e.g. the types of licenses under which software was made available, and the clauses they contain. These license clauses then may impose restrictions on which preservation actions can be performed on the software.
Patent information describes for example who is the owner of a specific patent, what the patent covers, or when it was granted. Patents also imply a restriction on how a software, hardware or method can be used.

While some parts of the Context Model of aprocess can be extracted automatically [30], especially in the aspects of dependencies and data formats, other aspects may still require significant manual work to obtain a proper representation. For example, the communication to a web service has to be described via the provision of its exact address and interface type. Similar situations apply to accessing databases, which are run as independent server processes.

V. Data Citation

Processes frequently process large volumes of data. To be able to repeat any such process we need to ensure that precisely the same sequence of data is fed as input into any such process. Storing a dump of such huge volumes of data, e.g. as part of the verification data in the context model, is not feasible in big data settings. We need to ensure that we can refer to the original data source / data repository for providing the data upon re-execution. While this may be rather trivial for static data sources being analyzed in their entirety, precise identification turns into a challenge in most settings where researchers use only a specific subset of entire data collection, and where this data collection is dynamic, i.e. subject to changes.

Most research datasets are not just static, but highly dynamic in their nature. New data is read from sensors or added from continuous experiments. Additional dynamics arises from the need of correcting errors in the data, removing erroneous data values, or re-calibrating and thus re-computing values at later points in time. Thus, researchers require a mechanism to retrieve a specific state of the data again, in order to compare the results of previous iterations of an experiment. Freezing the databases at specific points in time, batch-release of versions, etc. all provide rather inconvenient work-arounds, wasting storage space by keeping multiple copies of unchanged data in different releases, and delaying the release of new data by aggregating continous streams of data into batch releases.

Additionally, most processes will not analyze the entire database, but a very specific subset of it. We thus need to ensure that precisely teh same subset can be fed into the process again. Again, current approaches either waste space by storing explicit dumps of the subset used as input to a process, or require human intervention by providing (sometimes rather ambiguous) natural language descriptions of the subset of data used.

In order to address this issue, the Working Group on Dynamic Data Citation[8] (WGDC) of the Research Data Alliance (RDA) has devised a set of recommendations to address this challenge. In a nutshel, it relies on time-stamped and versioned storage of the data. Subsets are identfied by assigning persistent identifiers (PIDs) to time-stamped queries resolving to the subset. Hash-keys of the queries and the result sets are stored as metadata to allow verification of the resulting data sets upon re-execution [34], [35]. By shifting the focus from citing static data sets towards the citation of queries, which allow retrieving reproducible data sets from versioned data sources on demand, the problem of referencing acurate data sets can be addressed

[8]https://www.rd-alliance.org/groups/data-citation-wg.html

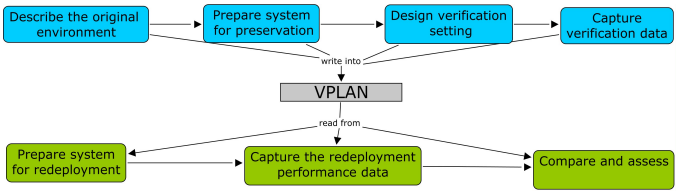


Fig. 3. VFramework used for verification of redeployed processes. [36]

more flexibly. It also provides additional provenance information on the data set by containing a semantic description of the subset in the form of filter parameters in the query. It furthermore allows retrieving the semantically identical data set including all corrections applied to it afterwards by re-executing the timestamped query with a later time-stamp. As the process can be automated it allows integrating data citation capabilities into existing workflows.

The persistent identifier serves as a handle which, in addition to representing the input of data in a specific process, can be shared with other peers and be used in publications. As the system is aware of updates and evolving data, researchers have transparent access to specific versions of data in their workflows. There is no need of storing multiple versions of a dataset externally for the long term as the system can reproduce them on demand. As hashing methods are in place, the integrity of the datasets can be verified. Thus the exact data set used during a specified workflow execution can be referenced by the PMP as explained in Section III.

VI. Verification and Validation

Upon re-executing a process (be it a simple reproduction or a repeatability setting after applying preservation actions, we need to verify the correct behavior in a potentially changed environment. The process of verification and validation (V&V) does not check the scientific correctness of the processes published by the researchers. It rather helps in obtaining evidence whether the replicated process has the same characteristics and performs in the same way as the original process.

In order to verify and validate the replicated process that was extracted from the source system and run in the target system, we follow the guidelines of [37] that describe the verification and validation of *transition activity*. We devised guidelines forming the VFramework [38] which are specifically tailored to processes and describe what conditions must be met and what actions need to be taken in order to compare the executions of two processes in different environments. The VFramework is presented in Figure 3 and consists of two sequences of actions.

The steps on the top are performed in the original environment, i.e. the system that a process is initially deplyoed in. The results obtained from the execution of each step are written into the VPlan. The second sequence depicted below is performed in the redeployment environment at any time in the future when the original platform may not be available anymore. Hence, it may be necessary to re-engineer the process in order to fit it into a new system. The necessary information is read from the VPlan. The measures obtained upon redeployment are compared with the measures from the original environment store din the VPlan using specific metrics (usually requiring them to be idenitcal or within certains tolerance intervals, depending on the significant properties of the process step/output to be compared).

There are a number of challenges that need to be taken into account during the V&V process. Some of the computational processes exchange data with external sources using a variety of network connections. These resources must also be available during the verification process so that the process can interact with them. A solution that allows monitoring of external services for changes, as well as their replacement for the purpose of verification and validation is described in [39].

Another challenge having influence on the verification is the lack of determinism of components. It can apply to both external resources that provide random values and to internal software components that, for example, depend on the system clock or the current CPU speed. In such cases the exact conditions must be re-created in both environments. Potentially, such components need to be substituted with their deterministic equivalents [12].

The Context Model contains information about dependencies required to run the software. If any of them was not identified with a use of automated tools or modelled manually, then the process will not execute. In the course of verification the Context Model gets improved until the process operates correctly. This is achieved either by repeating the capturing of the process using different process instances or by manual addition of identified process dependencies.

By verification and validation of the process automatically recreated in the target system we also indirectly verify and validate the Context Model. We prove its correctness and completeness, as the process could otherwise not be repeated and run correctly in the target system.

The methodology briefly described in here can be applied to verify and validate all cases in which the process is rerun, reproduced, or reused. In order to support the verification and validation for reproduction and reuse, it is important to also publish the verification data, because other researchers may not have access to the source system. Then they perform the verification and validation using the verification data provided by the experiment owner. This data should include significant properties, metrics and data extracted from the source system.

VII. Use Case

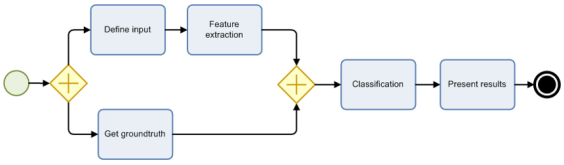


Fig. 4. Music Genre Classification Process [40]

We will use an example from the domain of music information retrieval (MIR) to illustrate the concepts presented in the preceeding sections. A common taks in is automatic classification of audio into some set of pre-defined categories, e.g. genres such as jazz, pop, rock, classic etc. at different levels of granularities. A process reflecting this task is depicted in Fig. 4. It requires the acquisition of both the actual audio files

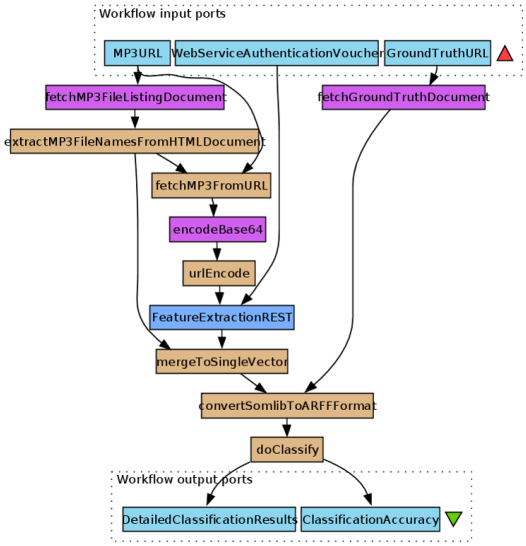


Fig. 5. Music Genre Classification Process modelled in Taverna

as well as ground truth information (i.e. pre-assigned genre labels for training and test data in the music collection) from some source. Next, some numeric descriptors (e.g. MFCCS, RhythmPatterns, SSDs) are extracted from the individual audio files via a range of signal processing routines and applying psycho-acoustic models to obtain feator vector representations of the audio. These are subsequently fed into some machine learning algorithm to train a classifier such as Support Vector Machines (SVM), Random Forest, and subsequently evaluated using performance measures such as recall and precision.

In one of our experiment settings this process was implemented using an (internally developed) web service for the feature extraction, WEKA as a third-party machine learning package, and a set of dedicated scripts and java applications for other tasks such as data acquisition, transformation, etc. These were orchestrated manually via the command line, or partially automated via shell scripts, all deployed on a linux system.

In order to increase repeatability and ease automatic analysis we migrated this process into a proper workflow representation using the Taverna workflow engine, as depicted in Fig. 5. It lists explicitly the data sources (URLs) where the audio files and ground truth labels are read from, as well as providing the authentication codes for the webservice that the audio files are sent to for feature extraction. The individual vector files are subsequently merged and fed into the classifier, which subsequently returns the actual classification results and the overall accuracy.

Applying a process monitoring tool we are able to autmatically capture all resources (files, ports) accessed or created by one instance of the process, depicted in Fig. 6. This includes, amongst others, a whole range of libraries (depicted in the upper left corner), the set of mp3 audio files (depicte din the lower left corner), a range of processes being called (e.g. wget to download the audio files and ground truth information, depcited in the upper right corner), the user id of the person calling the process, and others.

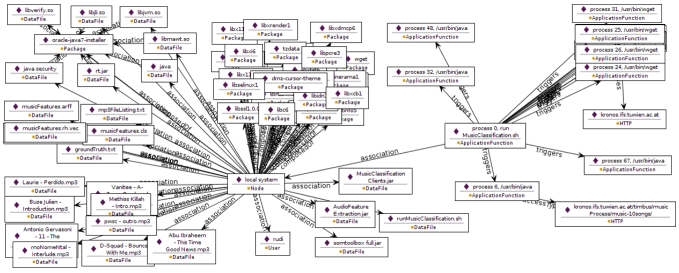


Fig. 6. Dependencies extracted from Music Genre Classification Process

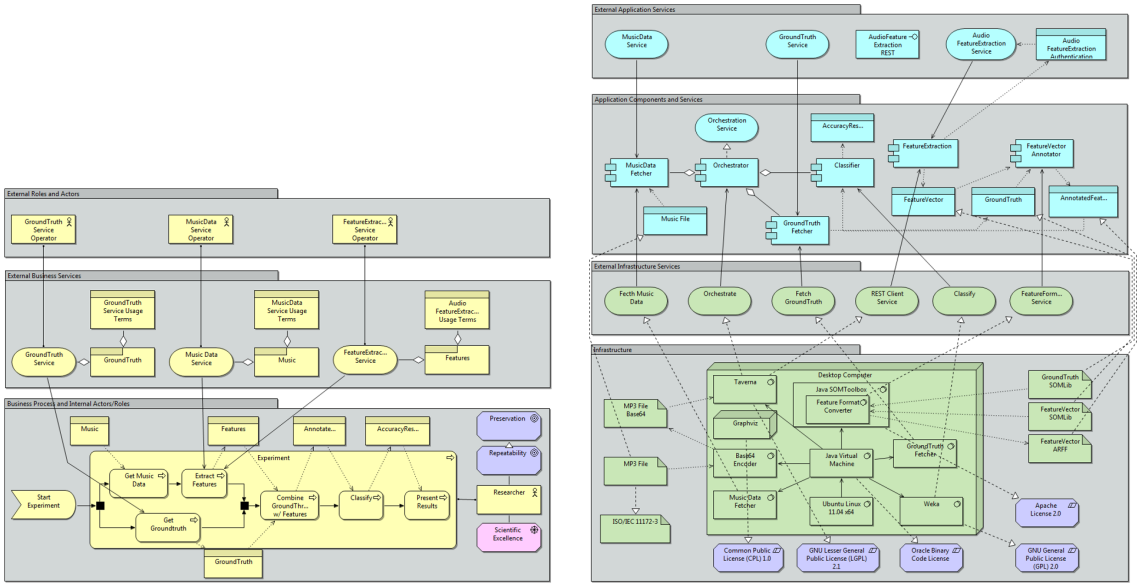
The raw information extracted bottom-up is subsequenlty enhanced, both automatically as well as manually, by structuring it according to the concepts provided by Archimate and adding aditional information, such as file format information being added by performing file format analysis using tools such as DROID, contacting file format registries such as PRONOM. The resulting structure is depicted in Fig. 7. Fig. 7a captures, at the bottom, the basic process and the objects (Music files, features extracted and passed on to the classifier, the ground truth annotations, and the final results). Stacked above it are the services being called, i.e. the audio feature extractor. In Fig. 7b the basic software (Java Virtual Machine, Weka, the data fetchers) are provided, with additional dependencies (e.g. the Unix Bash Shell, Base64 encoders, Ubuntu Linux in a specific version), with the data objects in different representations (e.g. the audio files as MP3 as well as base64-encoded MP3 files) and license information for the various tools (different versions of GPL, Apache Licene, Oracle Binary code License, the MP3 patent). On top of these, the detailed application components and services, both internal as well as external, are represented. This way, a comprehensive and well-structured documentation of the process can be obtained in a semi-automatic manner.

This information forms the Process Context Model and can be used for verification purposes. When applying the V-Framework, specific metrics are defined for comparing the input and output at each processing step. Upon re-executing the processin a different environment, these metrics are used to compare the results captured during re-execution, resulting in a verification report depicted in Fig.8

VIII. Conclusions and Future Work

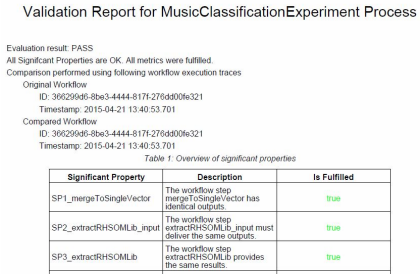
This paper describes a way to move beyond data-centric research by addressing the capture and description of entire research processes using Process Management Plans, that foster identification, description, sharing and preservation of scientific processes. We described the structure and contents of Process Management Plans and discussed the results of an analysis of various templates for Data Management Plans that identified their deficiencies with special attention put to support for processes. We also analysed how the key characteristics (traceability, repeatability, reproducibility, reusability, repurposeability, preservability) of modern science research are supported by Process Management Plans. We considered potential stakeholders for whom the information provided by the Process Management Plans would be useful.

In order to demonstrate how the core elements of Process Management Plans can be implemented we described how capturing of computational processes and their context can



(a) (b)

Fig. 7. Annotated Context Model of the Music Genre Classification Process



Validation Report for MusicClassificationExperiment Process

Evaluation result: PASS
All Significant Properties are OK. All metrics were fulfilled.
Comparison performed using following workflow execution traces
Original Workflow
ID: 366299d6-8be3-4444-817f-276dd00fe321
Timestamp: 2015-04-21 13:40:53.701
Compared Workflow
ID: 366299d6-8be3-4444-817f-276dd00fe321
Timestamp: 2015-04-21 13:40:53.701

Table 1: Overview of significant properties

Significant Property	Description	Is Fulfilled
SP1_mergeToSingleVector	The workflow step mergeToSingleVector has identical outputs.	true
SP2_extractRHSOMLib_input	The workflow step extractRHSOMLib_input must deliver the same outputs.	true
SP3_extractRHSOMLib	The workflow step extractRHSOMLib provides the same results.	true

Fig. 8. Verification of the Music Genre Classification Process (excerpt of report)

be performed. We also reviewed the recommendations of the Research Data Alliance on how to precisely identify arbitrary subsets of potentially high-volume and highly dynamic data. Last, but not least, we presented mechanisms for verification and validation of process re-executions.

The future work will focus on automation of PMP creation and verification by extraction of process characteristics automatically from its environment. We are currently evaluating the individual components of the PMP with stakeholders from different scientific communities. Specific focus is on tool support to automate many of the various documentation steps, specifically capturing and monitoring of low-level process characteristics and performance aspects. We incorporate all suggestions into a prototype implementation which fosters actionability and enforceability of Process Management Plans.

ACKNOWLEDGMENTS

This research was co-funded by COMET K1, FFG - Austrian Research Promotion Agency.

REFERENCES

[1] T. Hey, S. Tansley, and K. Tolle, Eds., *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Microsoft Research, 2009.

[2] E. Gronenschild, P. Habets, H. Jacobs, R. Mengelers, N. Rozendaal, J. van Os, and M. Marcelis, "The effects of freesurfer version, workstation type, and macintosh operating system version on anatomical volume and cortical thickness measurements," *PloS one*, vol. 7, no. 6, 2012.

[3] ISO, "ISO 5725:1:1994 Accuracy (trueness and precision) of measurement methods and results Part 1: General principles and definitions," ISO, Tech. Rep., December 1994.

[4] T. Miksa, S. Strodl, and A. Rauber, "Process management plans," *International Journal of Digital Curation*, vol. 9, no. 1, 2014.

[5] A. Curry, "Rescue of old data offers lesson for particle physicists," *Science*, vol. 331, no. 6018, pp. 694–695, 2011.

[6] M. Donnelly and S. Jones, "Checklist for a Data Management Plan," http://www.dcc.ac.uk/sites/default/files/documents/data-forum/documents/docs/DCC_Checklist_DMP_v3.pdf, 2011, accessed: 20/04/2013.

[7] Australian National Data Service, "ANDS Guides Awareness level - Data management planning," http://ands.org.au/guides/data-management-planning-awareness.pdf/, 2011, accessed: 20/04/2013.

[8] National Science Foundation, "Data Management for NSF EHR Directorate," www.nsf.gov/bfa/dias/policy/dmpdocs/ehr.pdf, 2011, accessed: 20/04/2013.

[9] K. Page, R. Palma, P. Holubowicz, G. Klyne, S. Soiland-Reyes, D. Cruickshank, R. G. Cabero, E. Garc'ia, D. D. R. Cuesta, and J. Zhao, "From workflows to research objects: an architecture for preserving the semantics of science," in *2nd International Workshop on Linked Science*, 2012.

[10] D. Garijo and Y. Gil, "A new approach for publishing workflows: Abstractions, standards, and linked data." in *6th workshop on Workflows in support of large-scale science*, 2011.

[11] R. Mayer, A. Rauber, M. A. Neumann, J. Thomson, and G. Antunes, "Preserving scientific processes from design to publication," in *Proceedings of the 16th International Conference on Theory and Practice of Digital Libraries (TPDL 2012)*. Springer, September 23–29 2012.

[12] M. Guttenbrunner and A. Rauber, "A measurement framework for evaluating emulators for digital preservation," *ACM Transactions on Information Systems (TOIS)*, vol. 30, no. 2, 3 2012.

[13] R. Darby, S. Lambert, B. Matthews, M. Wilson, K. Gitmans, S. Dallmeier-Tiessen, S. Mele, and J. Suhonen, "Enabling scientific data sharing and re-use," in *IEEE 8th International Conference on E-Science*, 2012.

[14] Charles Beagrie Ltd, "Economic impact evaluation of the economic and social data service," March 2011.

[15] D. De Roure, "Machines, methods and music: On the evolution of e-research," in *2011 International Conference on High Performance Computing and Simulation (HPCS)*, 2011, pp. 8–13.

[16] D. Roure, C. Goble, S. Aleksejevs, S. Bechhofer, J. Bhagat, D. Cruickshank, P. Fisher, N. Kollara, D. Michaelides, P. Missier, D. Newman, M. Ramsden, M. Roos, K. Wolstencroft, E. Zaluska, and J. Zhao, "The evolution of myexperiment," in *IEEE 6th International Conference on e-Science*, 2010, pp. 153–160.

[17] R. Mayer and A. Rauber, "A Quantitative Study on the Re-executability of Publicly Shared Scientific Workflows," in *Proceedings of the 11th IEEE International Conference on eScience*, 2015.

[18] P. Nowakowski, E. Ciepiela, D. Harezlak, J. Kocot, M. Kasztelnik, T. Bartynski, J. Meizner, G. Dyk, and M. Malawski, "The collage authoring environment," *Procedia CS*, vol. 4, pp. 608–617, 2011.

[19] C. Aiftimiei, A. Aimar, A. Ceccanti, M. Cecchi, A. D. Meglio, F. Estrella, P. Fuhrmam, E. Giorgio, B. Knya, L. Field, J. K. Nilsen, M. Riedel, and J. White, "Towards next generations of software for distributed infrastructures: The european middleware initiative." in *8th IEEE International Conference on E-Science*, 2012.

[20] T. Miksa, R. Mayer, S. Strodl, A. Rauber, R. Vieira, and G. Antunes, "Risk driven selection of preservation activities for increasing sustainability of open source systems and workflows," in *Proceedings of the 11th International Conference on Digital Preservation (iPres 2014)*, Melbourne, Australia, October 6–10 2014.

[21] H. Kulovits, C. Becker, M. Kraxner, F. Motlik, K. Stadler, and A. Rauber, "Plato: A preservation planning tool integrating preservation action services," in *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries 2008 (ECDL'08)*. Springer, 2008, pp. 413–414.

[22] U. B. Kejser, J. Davidson, D. Wang, S. Strodl, T. Miksa, K. H. E. Johansen, A. B. Nielsen, and A. Thirifays, "State of the art of cost and benefit models for digital curation," in *Proceedings of the Archiving Conference*, Berlin, Germany, 2014.

[23] T. Miksa, R. Mayer, M. Unterberger, and A. Rauber, "Resilient web services for timeless business processes," in *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services (iiWAS2014)*, Hanoi, Vietnam, December 4–6 2014, pp. 243–252.

[24] V. H. Publishing and A. Al, *Archimate 2.0: A Pocket Guide*, ser. TOGAF series. Van Haren Publishing, 2012.

[25] M. Lankhorst, *Enterprise architecture at work: modelling, communication, and analysis*. Springer, 2005.

[26] K. Belhajjame, O. Corcho, D. Garijo, et. al, "Workflow-centric research objects: First class citizens in scholarly discourse," in *Proceedings of Workshop on the Semantic Publishing, (SePublica 2012) 9th Extended Semantic Web Conference*, May 28 2012.

[27] H. Van de Sompel and C. Lagoze, "Interoperability for the Discovery, Use, and Re-Use of Units of Scholarly Communication," *CTWatch Quarterly*, vol. 3, no. 3, August 2007.

[28] P. Ciccarese, M. Ocana, L. Garcia Castro, S. Das, and T. Clark, "An open annotation ontology for science on web 3.0," *Journal of Biomedical Semantics*, vol. 2, no. Suppl 2, p. S4, 2011.

[29] D. De Roure, K. Belhajjame, P. Missier, J. Manuel, R. Palma, J. E. Ruiz, K. Hettne, M. Roos, G. Klyne, and C. Goble, "Towards the preservation of scientific workflows," in *Proceedings of the 8th International Conference on Preservation of Digital Objects*, Singapore, 2011.

[30] R. Mayer, G. Antunes, A. Caetano, M. Bakhshandeh, A. Rauber, and J. Borbinha, "Using ontologies to capture the semantics of a (business) process for digital preservation," *International Journal of Digital Libraries (IJDL)*, vol. 15, pp. 129–152, April 2015.

[31] W3C, "OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition)," Tech. Rep., December 2012, w3C Recommendation.

[32] R. Treinen and S. Zacchiroli, "Description of the CUDF Format," Tech. Rep., 2008, http://arxiv.org/abs/0811.3621.

[33] PREMIS Editorial Committee, "Premis data dictionary for preservation metadata," Tech. Rep., March 2008.

[34] S. Pröll and A. Rauber, "Data Citation in Dynamic, Large Databases: Model and Reference Implementation," in *IEEE International Conference on Big Data*, Santa Clara, CA, USA, October 2013.

[35] S. Proell and A. Rauber, "A Scalable Framework for Dynamic Data Citation of Arbitrary Structured Data," in *3rd International Conference on Data Management Technologies and Applications (DATA2014)*, Vienna, Austria, August 29-31 2014.

[36] T. Miksa, R. Vieira, J. Barateiro, and A. Rauber, "VPlan – ontology for collection of process verification data," in *Proceedings of the 11th International Conference on Digital Preservation (iPres 2014)*, Melbourne, Australia, October 6–10 2014.

[37] "IEEE Std 1012 - 2012 IEEE Standard for Software Verification and Validation," Tech. Rep., 2012.

[38] T. Miksa, S. Proell, R. Mayer, S. Strodl, R. Vieira, J. Barateiro, and A. Rauber, "Framework for verification of preserved and redeployed processess," in *10th Conference on Preservation of Digital Objects (IPRES)*, 2013.

[39] T. Miksa, R. Mayer, and A. Rauber, "Ensuring sustainability of web services dependent processes," *International Journal of Computational Science and Engineering (IJCSE)*, vol. 10, no. 1/2, pp. 70–81, 2015.

[40] R. Mayer and A. Rauber, "Towards time-resilient mir processes," in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, October 2012.