Minimal Sufficient Information about the Scientific Workflows to Create Reproducible Experiment

Anna Bánáti[3], Péter Kacsuk[1,2], Miklós Kozlovszky[1,3]
[1] MTA SZTAKI, H-1518 Budapest, Pf. 63., Hungary
[2] University of Westminster, 115 New Cavendish Street, London W1W 6UW
[3] Óbuda University, John von Neumann Faculty of Informatics, Biotech Lab
Bécsi str. 96/b., H-1034, Budapest, Hungary
peter.kacsuk@sztaki.mta.hu
{banati.anna, kozlovszky.miklos}@nik.uni-obuda.hu

***Abstract* — The reproducibility of an in-silico experiment is a great challenge because of the parallel and distributed environment and the complexity of the scientific workflows. In order to solve such problems on one hand provenance data has to be captured about the dataflow, the ancestry of the results and the environment of the execution, on the other hand description data has to be collected from the scientist and stored about the essential details, the types and samples of input/output data, and the operation of the experiment. The ultimate goal of our work is to propose a minimal dataset for recording and reporting scientific workflow based experiment, which will facilitate the reproducibility of such experiments, the public repositories and enable to share and reuse the scientific result. One part of the dataset can be filled in manually by the scientist, certain part can be filled in automatically by the system and other part can be filled in from provenance data.**

I. INTRODUCTION

In large computational challenges (scientific) workflows have emerged as a widely accepted solution for performing in-silico experiments. In general these in-silico experiments consist of series of particularly data and compute intensive jobs, (called scientific workflow), and in most cases their executions require parallel and distributed infrastructure (super/hypercomputers, grids, clusters, clouds).

An essential part of the scientific method is that researchers can repeat and reproduce the experiments of others and test the outcomes themselves even in a different environment. Different users for different purposes may be interested in reproducing the workflow, for example the authors of workflow in order to prove their results, readers or other scientists in order to reuse results or reviewers in order to verify the correctness of the results [1]. Additionally, nowadays scientific workflow repositories are available and in this way the scientists can share their results with each other and even they can reuse the existing workflows to create new ones.

The implementation of the reproducible and reusable scientific workflows is not an easy task and many obstacles have to be removed toward the goal. Three main components play important role:

1. The scientific workflow management system (SWfMS) should support the scientist with automatic provenance data collection about the environment of execution and about the data production process. In our previous work [2] we determined the four level of the provenance, and the different utilizations of the captured data in the different levels. Capturing provenance data during the running time of the workflow is crucial to create reproducible workflows.

2. The scientists should carefully design the workflow (for example with special attention for modularity and robustness of the code [3]) and give a description about the operation of experiment, the input and output data, even they should show samples. [4], [5].

3. The dependencies of the workflow execution should be eliminated. A workflow execution may depend on volatile third party resources and services; special hardware or software elements which are available only in a few and special infrastructure; deadlines, which cannot be accomplished on every infrastructure or it can be based on non-deterministic computation which apply for example random generated values [2].

Our goal is to support and facilitate the work of the scientist by the scientific workflow management system (SWfMS) to create a well-documented and reproducible scientific workflow. The basic idea of our work is given by MIAME which describes the Minimum Information About a Microarray Experiment that is needed to enable the interpretation of the results of the experiment unambiguously and potentially to reproduce the experiment. [6], [7]. We collected and categorized the minimal sufficient information into seven different datasets, which target different problems to solve. Accordingly, one of the types of data serves the documentation of experiment and helps to share it in a scientific workflow repository. Other type of data describes the data dependency and the process of data product and it is necessary for the proving and verification of the workflow. There are data which are needed to the repeatability or reproducibility of workflows in different infrastructure and environment. Finally we collected information to help identifying the critical points of the execution which reduce the possibility of reproducibility or even arrest it.

The datasets are created in the different phases of the scientific workflow lifecycle [8], [9] and originate from three different sources. The scientist can give information when to design the abstract model, when to get the results

or after the results are published. Other information can be gained from provenance database and there is information which can be generated automatically by the system.

With the help of our proposal we wish to solve the following problems:

- how to create a detailed description about scientific experiment;
- which minimal information is necessary to be collected from the scientists about their experiments to achieve a reproducible workflow;
- which minimal information is necessary from provenance to reproduce the experiments;
- which data and information can be generated automatically by the SWfMS in order to implement a reproducible scientific workflow;
- which jobs at which point do not meet the requirements of independencies.

The rest of the paper is organized as follows. In the next section, we present the background of the workflow reproducibility and the connected work of the research field. In Section 3. We define the seven datasets and give an overview of their purposes. The next two section deal with the datasets belonged to the jobs and their dependencies and finally we summarize our conclusions and reveal the possibility of future research direction.

II. BACKGROUND AND RELATED WORKS

Currently the reproducibility of scientific workflows is a burning problem which the scientists and the system developers have to face with and have to find solutions. Many researchers investigate this issue, analyze the requirements of reproducibility and deal with the implementation of tools or frameworks which facilitates reproducibility of the workflow.

The researchers agree on the importance of the careful design [10], [11], [12], [13], [14], for example the modular design and programming, the detailed description of the workflow, the input/output data examples, and consequent annotations [3]. In addition the careful design includes the careful usage of volatile third parties or special local services. In these cases two solutions exist, but

TABLE I.
OVERVIEW TABLE ABOUT THE NECESSARY DATASETS

	filled in by the scientist	filled in from Provenance db.	automatically generated by the system
general description of experiment	title, topic, author(s), date, institute, laboratory, description, publication details, experiences, comment	number of ex-submission, number of failure, duration of execution, statistical data based on previous execution	workflow ID,
detailed description of workflow	abstract wf model (DAG) , wf version, parents, used parameter set, requirements (resources, libraries, applications with version number), place of input/output data files or storage), types of input/output data, constraints, deadlines, dependencies, etc..	wf version, parents, statistical data about previous execution, timestamps, resource usage, failure rate, etc..	num of job, num of i/o port; num of entry/exit job
detailed description of infrastructure	infrastructure, OS, middleware, required resources, number of VM, etc.		
detailed description of environment	authentication parameters, required libraries, compilers, functions,	start/end time of execution, statistical data based on the actual or previous execution, resource usage (CPU, RAM, DISK, stb),	
detailed description of job	input/output data, types of input/output data, volume of input/output data, example input/output data, place of input/output data, required application and its details, version number of app., dependencies, constrain, etc..	parents, statistical data about previous execution,	num of i/o ports, predecessors, successors, etc..
detailed description of the environment of the job	type of code,	time stamps (exec&wait time), resource usage, failure rate, etc..	compiler, required libraries
dependency dataset	it is automatically generated by the system based on the response of scientist		

reproducibility is uninsurable: 1. taking a digital copy of the entire environment using a system virtual machine/hardware virtualization approach 2. capturing and storing metadata about the code and environment that allows it to be recreated later [3]. In [4], [5], [15] the authors give further "best practice" and draw attention for the phenomena of *workflow decay* [4], which means that year by year the ability and success of the re-execution of any workflow significantly reduces.

Consequently we can declare that the reasons are revealed from the problem but the solution is not trivial, cannot be implemented in every cases and most of all the workflow management systems do not force yet the user to make a reproducible workflow.

VisTrail, ReproZip or PROB [16], [17], [18] are all available tools that assist the researchers and scientist to create reproducible workflows. VisTrail [16], [19] provides help for creating detailed descriptions not only about the scientific experiment but also about the links for input data, applications and visualized output which always harmonizes with the actually applied input data, filter or other parameters while ReproZip [17] creates a self-contained reproducible package by stitching together the detailed provenance information and the environmental parameters. These tools can be used in many cases, but do not pay attention for example the volatile third party services or non-deterministic applications.

Currently the Research Object (RO) approach [20] is the main direction in this research field. RO defines an extendable model, which aggregates a number of resources in a core or unit, namely: a workflow template; workflow runs obtained by enacting the workflow template; other artifacts which can be of different kinds; annotations describing the aforementioned elements and their relationships. Accordingly to RO the authors in [21] also investigate the requirement of reproducibility and the required information to achieve it. They created ontologies, which help to uniform these data. These ontologies can help our work too in order to implement a more general solution.

Gesing at al. in [22] describe the approach targeting various workflow systems and building a single user interface for editing and monitoring workflows under consideration of aspects such as optimization and provenance of data. Their goal is to ease the use of workflows for scientists and other researchers. They designed a new user interface and its supporting infrastructure which makes it possible to discover existing workflows, modifying them as necessary, and to execute them in a flexible, scalable manner on diverse underlying workflow engines.

III. DATASETS

We defined seven types of datasets which contain the necessary and sufficient information about the experiment. An overview table summarizes the seven datasets and shows some examples about the stored data. (Table 1.) Data collected into different datasets target different problems to solve.

We present one sample table of the seven datasets about the Detailed Description of Environmental of Job in Appendix A. We highlighted the rows which can affect the reproducibility of the workflow.

One part of the collected information of these datasets originates from the user, who creates the workflow. In the design phase the user establishes the abstract workflow model, defines the jobs, determines the input/output ports, specifies the input data and so on. Simultaneously, in order to achieve the reproducibility of workflow the user has to create the appropriate documentation about the experiment in a specific way, form and order. Such information is for example some personal data (name, date, etc), the description of experiment (title, topic, goal, etc.), the samples about the necessary input, partial and output data, special hardware, application or service requirements and so on.

There are provenance data too in the datasets which have to be captured by the SWfMS in running time. For example the version number and the variation of a given workflow, the number of submissions, the used data or parameter set during the previous executions, the makespan of execution or the number and types of failures occurred in running time. Information like these can be also crucial when the results of experiment have to be reproduced in a later time or in a different environment.

The third type of information is generated automatically by the system after the workflow is submitted, in the instantiation phase of the workflow lifecycle. This information can be obtained from the users too, but simpler, faster and even more precise and trusty if it is automated (for example workflow and job IDs, number of ports etc). There exists such information too, which is created manually by the user at the beginning, but since the datasets and the database continuously grow and more and more data are collected, the system could "learn" certain information and fill in automatically the appropriate entries of datasets.

A. General Description of Workflow (GDW).

This dataset contains general information about the scientific experiment such as title; author's name and its profile; the date; the institute's name and address, where the experiment is conducted and so on. In addition, general description of the experiment and data samples is also very important to be documented and stored. Most of the information originated from the users and it is necessary to create well-documented workflows, which will be reusable and understandable even after years. Certain entries are created in the design phase and others after the execution or later (for example publication details). However there exist information which is generated automatically by the SWfMS, such as Experiment ID, which is a unique identifier (expID) referred to the given workflow.

B. Detailed Description of Workflow (DDW)

The specification of the workflow is stored in the DDW. The experiment is modelled with an acyclic directed graph (DAG) (figure 1.) which is the most important part of this documentation in a graphical manner too. In addition detailed information can be found in this dataset about the workflow (version number, parent workflows, required parameter set), the input/output data (number, type, amount, location, access method) the optional constraints or deadlines or other requirements. Automatically generated information is for example the number of input/output ports, the number of jobs, the number of entry/exit tasks

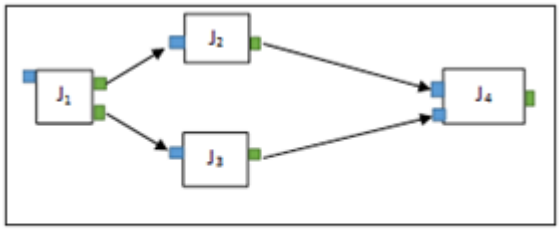

Figure 1. Scientific Workflow example with four jobs (J1, J2, J3, J4)

C. Detailed Description of Infrastructure (DDI).

If the goal is to repeat or reproduce the workflow execution on a different infrastructure, we have to store the descriptors and parameters of the infrastructure, the middleware and the operating systems in details too.

D. Detailed Description of Environment (DDE).

If the goal is to repeat or reproduce the workflow execution in a later time, we have to store the detailed environmental parameters. In this dataset the following data can be found: the environmental variables and parameters; the circumstances of the execution; the state descriptors of the used resources; the time stamps; the required libraries, applications, data and services (with their exhaustive descriptions such as location, access method, version number etc.). This information can be captured during execution and can be stored as provenance data in a provenance database. The fields of this dataset filled in from this database.

IV. DATESETS FOR JOBS

Every job has two datasets, the Detailed Description of Job (DDJ) and the Detailed Description of Environment of Job (DDEJ). Data in DDJ was collected on the basis of two aspects: the first one helps understand the operation of a given job. The second one helps follow the computational process and partial or final results. DDEJ stores information about the environmental parameters of the execution, which serves the reproducibility. The number of DDJs (and also DDEJ) is equal to the number of jobs in the whole workflow.

A. Detailed Description of Job (DDJ)

The jobs in the abstract workflow model are organized into levels. The predecessors of any job are in lower level, the successors of a job are in upper level. This precedency appears in the naming convention of the job ID, which is referred to the exp ID and the sequence number of a level and the sequence number of a job in the given level. The entry job has not any input port or predecessor job, the exit job has not any output port or successor job.

Also in this case certain entries originate from the user (general description, job's name, sample input/output data, location and access method of input/output data, special hardware/application/service requirements etc.) and others are generated automatically by the system (job ID, predecessor and successor jobs, number of input/output ports, resource requirements).

B. Detailed Description of Environment of Job (DDEJ)

Provenance data can be used to fill in the most fields, such as type and number of failures; failure rate; start/end time of execution, waiting time, used resources, statistical data about previous executions and so on. The rest of necessary information can be generated automatically by the SWfMS such as type of code, compiler, resource requirements, virtual machine requirements and its state descriptors and so on.

V. DEPENDENCY DATASET

In the instantiation phase of the workflow lifecycle, the SWfMS can examine the dependencies of the submitted workflow. With help of the given results together with the information gained from the user the system can create a so called Dependency Dataset, which will store all the jobs which depend on any external circumstances and may not be reproducible. In our previous paper [2] we showed, that the rate of reproducibility of a scientific workflow can be computed with the help of which the reproducible parts of workflow can be determined. From this dataset, after viewing the results the user – before finally submits his workflow – can think over the model, he can modify it and can eliminate certain dependencies or he can decide to apply extra provenance or virtualization tools to preserve the workflow.

VI. CONCLUSIONS AND FUTURE WORKS

In this paper we investigated the necessary and sufficient information about scientific workflows to make them reproducible. We defined seven minimal datasets to achieve our goal. These datasets target the documentation of the experiment, the verification of workflows, the reproducibility and the reusability of workflows. The datasets - related to the whole workflow and to the particular jobs - are filled in from three different sources: the scientist, the system and the provenance database. These datasets among others contain detailed information about the operation of the experiment; description and samples about input, partial and output data; and environmental descriptors. In addition we specified another dataset about jobs depending on external conditions or non-deterministic factors, which can affect or even prevent the reproducibility or reusability of workflows. Based on this dataset our goal is to determine the probability of reproducing workflow whether in a later time it will give the same results.

The goal of the defined datasets is to propose a general solution to support the user by the SWfMSs in creating reproducible workflows. The dashboard approach described in [22] aims to convince the researchers to start using workflows extensively hiding the technical aspect of workflows. Our future work is to support this concept with our minimal sufficient information concept helping the scientist to create reproducible workflow in an easy way.

REFERENCES

[1] D. Koop, E. Santos, P. Mates, T.Vo Huy, P Bonnet, B. Bauer, M.Troyer, D.N. Williams, J.E. Tohline, J. Freire, C.T. Silva, „A Provenance-Based Infrastructure to Support the Life Cycle of Executable Papers", Internatioonal Conference on Computational Science, ICCS 2011.

[2] A. Banati, P. Kacsuk, M. Kozlovszky, M. Four level provenance support to achieve portable reproducibility of scientific workflows. In Information and Communication Technology,

Electronics and Microelectronics (MIPRO), 2015 38th International Convention on (pp. 241-244). IEEE.

[3] A. Davison, „Automated Capture of Experiment Context for Easier Reproducibility in Computational Research”, Computing in Science & Engineering, vol 14/ 4, pp. 48–56, july. 2012

[4] J. Zhao, J. M. Gomez-Perez, K. Belhajjame, G. Klyne, E. Garcia-Cuesta, A. Garrido, K. Hettne, M. Roos, D. De Roure, és C. Goble, „Why workflows break—Understanding and combating decay in Taverna workflows”, in E-Science (e-Science), 2012 IEEE 8th International Conference on, 2012, o. 1–9.

[5] K. M. Hettne, K. Wolstencroft, K. Belhajjame, C. A. Goble, E. Mina, H. Dharuri, D. De Roure, L. Verdes-Montenegro, J. Garrido, és M. Roos, „Best Practices for Workflow Design: How to Prevent Workflow Decay.”, in SWAT4LS, 2012

[6] http://fged.org/projects/miame/

[7] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, M. Vingron, M. Minimum information about a microarray experiment (MIAME)—toward standards for microarray data. Nature genetics, 29(4), 365-371., 2011

[8] E. Kail, A. Bánáti, K. Karóczkai P. Kacsuk, M. Kozlovszky, Dynamic workflow support in gUSE. In Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on (pp. 354-359). IEEE.

[9] B. Ludäscher, I. Altintas, S. Bowers, J. Cummings, T. Critchlow, E. Deelman, D. D. Roure, J. Freire, C. Goble, és M. Jones, „Scientific process automation and workflow management”, Scientific Data Management: Challenges, Existing Technology, and Deployment, Computational Science Series, o 476–508, 2009

[10] P. Missier, S. Woodman, H. Hiden, és P. Watson, „Provenance and data differencing for workflow reproducibility analysis”, Concurrency and Computation: Practice and Experience, 2013

[11] R. D. Peng, „Reproducible Research in Computational Science”, Science, köt. 334, sz. 6060, o. 1226–1227, dec. 2011

[12] J. P. Mesirov, „Accessible Reproducible Research”, *Science*, köt. 327, sz. 5964, o. 415–416, jan. 2010.

[13] D. De Roure, K. Belhajjame, P. Missier, J. M. Gómez-Pérez, R. Palma, J. E. Ruiz, K. Hettne, M. Roos, G. Klyne, C. Goble, és others, „Towards the preservation of scientific workflows”, in Procs. of the 8th International Conference on Preservation of Digital Objects (iPRES 2011). ACM, 2011.

[14] S. Woodman, H. Hiden, P. Watson, és P. Missier, „Achieving reproducibility by combining provenance with service and workflow versioning”, in Proceedings of the 6th workshop on Workflows in support of large-scale science, 2011, o. 127–136.

[15] P. Groth, E. Deelman, G. Juve, G. Mehta, és B. Berriman, „Pipeline-centric provenance model”, in Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science, 2009, o. 4

[16] J. Freire, D. Koop, F. S. Chirigati, és C. T. Silva, „Reproducibility Using VisTrails”, 2014. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download doi:10.1.1.369.9566

[17] F. S. Chirigati, D. Shasha, és J. Freire, „ReproZip: Using Provenance to Support Computational Reproducibility.”, in *TaPP*, 2013.

[18] V. Korolev, A. Joshi, V. Korolev, M. A. Grasso, A. Joshi, M. A. Grasso, D. Dalvi, S. Das, V. Korolev, Y. Yesha, és others, „PROB: A tool for Tracking Provenance and Reproducibility of Big Data Experiments.”, *Reproduce’14. HPCA 2014*, köt. 11, o. 264–286, 2014.

[19] D. Koop, J. Freire, és C. T. Silva, „Enabling Reproducible Science with VisTrails”, *arXiv preprint arXiv:1309.1784*J. Cheney, A. Finkelstein, B. Ludäscher, és S. Vansummeren, „Principles of provenance (dagstuhl seminar 12091)”, Dagstuhl Reports, köt. 2, sz. 2, 2012

[20] O. Belhajjame, K. Corcho, D. Garijo, J. Zhao, P. Missier, D. R. Newman, R. Palma, S. Bechhofer, G. C. Esteban, J. M. Gomez-Perez, G. Klyne, K. Page, M. Roos, J. E. Ruiz, 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 the ESWC2012 Workshop on the Future of Scholarly Communication in the Semantic Web, 2012.

[21] K. Belhajjame, J. Zhao, D. Garijo, M. Gamble, K. Hettne, R. Palma, E. Mina, O. Corcho, J. M. Gómez-Pérez, S. Bechhofer G. Klyne C. Goble, Using a suite of ontologies for preserving workflow-centric research objects. Web Semantics: Science, Services and Agents on the World Wide Web. 2015

APPENDIX A
DETAILED ENVIRONMENTAL DESCRIPTION OF A JOB (DEDJ)

type of code	Ontologi term (OT)	
compiler	OT	opt auto
number of necessary library	text	the next fields depend on answer
location of lib1	text	
required application	OT	the next fields depend on answer
location of app (access path)	text	
access method of app	OT	
number of input port	text	automatically, the next fields depend on answer
arrival time of input on port1	text	from prov
amount of received data on port1	text	from prov
transfer time on this edge	text	from prov
transfer method	OT	
CPU requirements	text	from prov
RAM requirements	text	from prov
Disk requirements	text	from prov
number of VM requirements	text	the next fields depend on answer
Type of VM1	OT	
OS on VM1	OT	
CPU requirements of VM1	text	
RAM requirements of VM1	text	
Disk requirements of VM1	text	
Number of Application on VM1	text	the next fields depend on answer
number of special hardware demand	text	the next fields depend on answer
type of special hardware1	OT	
method of access this hw	OT	
type of authentication to access this hw	OT	
number of third party service	OT	the next fields depend on answer
type of this service	OT	
method of access this service	OT	
type of authentication to this service		
third party data demand	yes/no	
method of access this data		
amount of this data		
execution time (makespan)		from prov
start time		from prov
waiting time in a queue		from prov
deadline of execution	text	

[22] S. Gesing, M. Atkinson, R. Filgueira, I. Taylor, A. Jones, W. Stankovski, C. S. Liew, A. Spinuso, G. Terstyanszky and P. Kacsuk, Workflows in a dashboard: a new generation of usability. In proceedings of 9th Workshop on Workflows in Support of Large-Scale Science (WORKS), pp. 82-93. IEEE. 20144