Distilling Structure in Scientific Workflows

Jiuqiang Chen, Christine Froidevaux, Carole Goble, Alan Williams, Sarah Cohen-Boulakia

DISTILLING STRUCTURE IN SCIENTIFIC WORKFLOWS

Jiuqiang Chen1,2,3, Christine Froidevaux[1,2], Carole Goble[4], Alan R Williams[4]
Sarah Cohen-Boulakia[1,2]

[1] Laboratoire de Recherche en Informatique, CNRS UMR 8623, Bât 650, Université Paris Sud, 91405 Orsay Cedex France
[2] AMIB group, INRIA Saclay, France
[3] School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China
[4] School of Computer Science, The University of Manchester, Oxford Rd., Manchester, UK
e-mail: chenj@lri.fr, chris@lri.fr, carole.goble@manchester.ac.uk, alanrw@cs.man.ac.uk, cohen@lri.fr (corresponding)

Motivation and Objectives

Scientific workflows management systems, (*e.g.,* (Missier et al., 2010; Ludaescher et al., 2006; Goeck et al. 2011)) are increasingly used to specify and manage bioinformatics experiments. An experiment is then represented by a workflow in which a large number of bioinformatics tasks are linked to each other. A *workflow specification* is a framework for the execution of workflows. It specifies the order to be observed between the different tasks and their relationships with the workflow inputs and workflow outputs. According to the input data given to the workflow specification and assignments of values to the task parameters, different *workflow runs* are then obtained and may lead to different intermediate and final output data. Both workflow specifications and runs are represented by graphs.

Faced with the increasing complexity of runs and the need for reproducibility of results, provenance has become an important research topic. A significant number of tools for managing vast amounts of data provenance have been designed to assist the storage of provenance data (*e.g.*, indexing), query the data (*e.g.*, difference between executions, search for patterns), visualize the workflow provenance or (re)schedule executions... (See (Cohen-Boulakia and Leser, 2011) for a review on that topic). These tools all make intrinsically complex operations on graph structures (search for subgraphs in a graph, comparing graphs, ...), which, if carried out on Directed Acyclic Graphs (DAGs), with no other restriction of structure, lead to NP-hard problems. Instead, these problems can be solved in polynomial time when specific restrictions are imposed on graphs, such as considering *series-parallel* (SP) structures (Bein et al., 1992). Some provenance management approaches such as (Bao et al., 2009; Callahan et al., 2006) have therefore chosen to restrict workflow graphs to SP structures. As in general, workflows obtained using workflow systems are DAGs with any structure, graphs transformation approaches such as (Escribano et al., 2009) can be exploited to transform workflow graphs into SP graphs. (Cohen-Boulakia et al, 2012) has recently designed SPFlow, the first algorithm able to rewrite any scientific workflow graph structure into an SP workflow structure while preserving provenance information. As expected, such an approach has a cost in that nodes and/or edges have to be duplicated in the rewritten workflow.

Determining the reasons why some workflows have non SP structures may help users to directly design workflows having a structure closer to SP structures. The rewriting process may then be used on less complex, *distilled,* workflows. The aim of this paper is to present the results obtained on the study that we have conducted on the set of Taverna workflows (Missier et al., 2010) available on myExperiment (De Roure et al, 2009) to analyze the reasons why workflows have non SP structures.

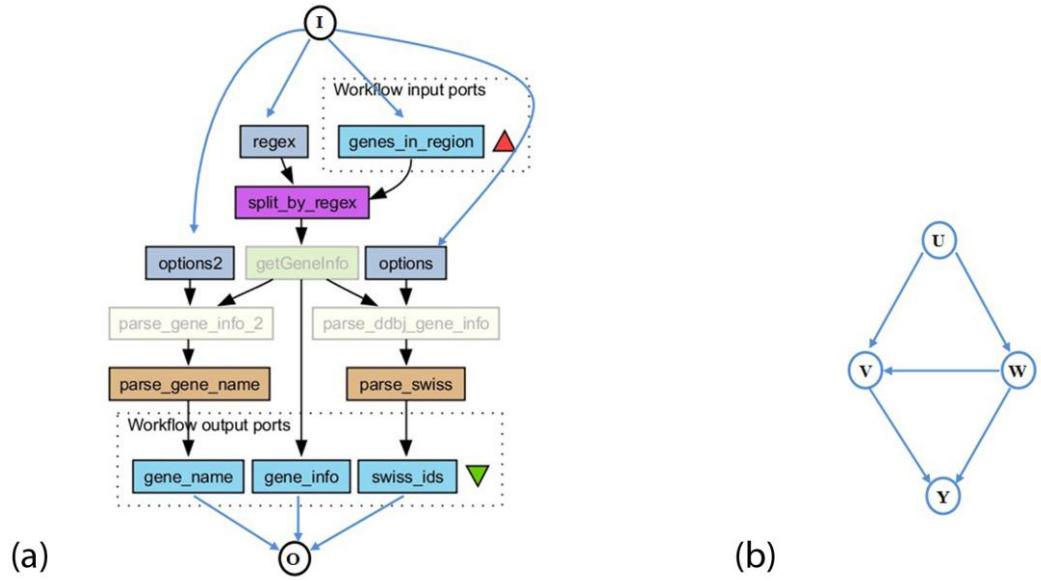

Figure 1: (a) Example of a non SP structure from myExperiment; (b) forbidden pattern

Methods

Our study has been conducted on a set of 1,014 distinct workflows extracted from the Taverna workflows available in myExperiment in May 2012. We have implemented the algorithm of (Valdes et al., 1979) to detect whether workflow graphs are SP. Intuitively, SP structures are graph structures having one main input (I in figure 1(a)) and one main output (O in Figure 1(a)), without loops and which can be synchronized. In particular the pattern highlighted in Figure 1 (b) is forbidden (in this pattern, arcs can be replaced by paths involving intermediate nodes). In this pattern, node w is responsible for the graph non to be SP. Such a node is called a *reduction node* (Bein et al., 1992) and is duplicated in SPFlow. In the workflow depicted in Figure 1(a) the *getGeneInfo* processor is a reduction node so that the workflow is not SP. Among the 390 workflows with non SP structures (38,5%), we have focused on identifying reduction nodes and analyzed the forbidden pattern in which they were involved.

We have then driven two series of experiments:
- The first series of experiments has consisted in analyzing the structure of a subset of workflows having complex non SP structures.
- In the second series of experiments, we have considered all the non SP workflows of Taverna and we have conducted a study of the processors involved in non SP structures. We have identified the kinds of processors mostly involved in non SP structures and we have then made a more precise analysis by examining the processors themselves.

Results and Discussion

Trace links: The first series of experiments highlight the fact that some intermediate processors are directly linked to the workflow outputs merely for the sake of keeping track of intermediate results. We call such intermediate processors *trace nodes* and their outgoing edges linked to the workflow outputs are called *trace links*. On a total of 13,754 nodes in the set of non SP workflows, we found 1,524 reduction nodes including 631 reduction nodes that are also trace nodes (representing 41% of the reduction nodes) and involved in 361 workflows (representing 92.6% of non SP workflows).

Interestingly, trace links could be removed by exploiting the powerfulness of the provenance module of Taverna that is in charge of collecting all intermediate and final results obtained and consumed during each execution.

Ongoing work includes focusing on the workflows for the *BioVeL* project and work in close collaboration with the workflow writers for potential improvement in the structure of some workflows when trace links may appear.

Non-SP-only processors: The second series of experiments revealed that most reduction nodes correspond to local processors (processors provided by Taverna to workflow designers) and web services processors. In particular, among a set of 92 web services, 40 services only appear in non SP workflows and occur at least once as reduction nodes. More interestingly, nine services appear only as reduction nodes in Non SP workflows. We call them *Non-SP-only processors*. As for local services, we found one Non-SP-only local processor.

Ongoing work includes investigating ways to modify the use of Non-SP-only processors (*e.g.*, changing the processors ports, grouping several consecutive calls of the same processor, designing SP patterns of joint use) so that they are not anymore systematically associated to (and possibly responsible for) non SP structures.

In conclusion, we have identified several reasons why workflows may not have an SP structure. Following the solutions underlined, we will get *distilled* workflows in which the number of reduction nodes should importantly be reduced and we hope that a large part of workflows may become SP. In our approach, users do not have to consider structural constraints when they design workflows; our aim is instead to provide them with designing guidelines ensuring that *distilled* workflows are naturally produced.

Acknowledgements

This work has been partly supported by the INRA-INRIA ASAM project.
J. Chen has been supported by the China Scholarship Council (CSC).

References

- Bao Z, Cohen-Boulakia S, Davidson SB, Eyal A, Khanna S. (2009) Differencing provenance in scientific workflows, *Proc. of ICDE 2009, 808-819*. doi: 10.1109/ICDE.2009.103
- Bein W, Kamburowski J, Tallmann MF. (1992) Optimal reductions of two-terminal directed acyclic graphs, *SIAM J. Comput.* 21(6):1112--1129. dio: 10.1137/0221065
- Callahan SP, Freire J, Santos E, Scheidegger CE, Silva CT et al. (2006) Vistrails: visualization meets data management, *Proc. of SIGMOD 2006, 745-747*. doi: 10.1145/1142473.1142574
- Cohen-Boulakia S, Froidevaux C, Chen J. (2012) Scientific Workflow Rewriting while Preserving Provenance. *Proc of the 8th IEEE Int. Conference on eScience*. (In press)
- Cohen-Boulakia S, Leser U (2011) Search, adapt, and reuse: the future of scientific workflows, *SIGMOD Record 40(2):6-16*. dio: 10.1145/2034863.2034865
- De Roure D, Goble C, Bhagat J, Cruickshank D, Goderis A et al. (2009) The Design and Realisation of the myExperiment Virtual Research Environment for Social Sharing of Workflows. *Future Generation Computer Systems 25:561-567*. doi: 10.1109/eScience.2008.86
- Escribano A, van Gemund A J.C, Cardeñoso-Payo V. (2009) Performance implications of synchronization structure in parallel programming, *Parallel Computing 35(8-9) 455-474*. dio: 10.1016/j.parco.2009.07.002
- Goecks J, Nekrutenko A, Taylor J, The Galaxy Team. (2011) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biology 11(8):R86*. doi:10.1186/gb-2010-11-8-r86
- Ludaescher B, Altintas I, Berkley C, Higgins D, Jaeger E et al. (2006) Scientific workflow management and the Kepler system. Concurrency and Computation: *Practice and Experience 18(10):1039-1065*. doi: 10.1002/cpe.994
- Missier P, Soiland-Reyes S, Owen S, Tan W, Nenadic A et al. (2010) Taverna, Reloaded. *SSDBM 2010, 471-481*. DOI: 10.1007/978-3-642-13818-8_33
- Valdes J, Tarjan RE, L. Lawler E. (1979) The recognition of series parallel digraphs, *STOC, 1—12*. doi: 10.1145/800135.804393