Towards a Formal Foundation for Aggregating Scientific Workflows

Frank Terpstra, Zhiming Zhao, Wico Mulder, and Pieter Adriaans

Informatics Institute
University of Amsterdam
Kruislaan 419, 1098VA, Amsterdam, The Netherlands
{ftrpstra,zhiming,wicomul,pietera}@science.uva.nl

Abstract. In e-Science, scientific workflow systems are used to share data and knowledge in collaborative experiments. In recent work we discussed the concepts of a workflow bus [1], allowing multiple workflow systems to be coupled in a meta-workflow system with multiple execution models. In this paper we propose an approach for a formal model to perform the task of reasoning of about the execution models of such workflow systems. We propose that I/O Automata can be used as a formalism to prove the correctness of complicated workflows involving multiple workflow engines and execution models.

Keywords: I/O Automata, formalism, workflow design.

1 Introduction

In scientific workflow research many different implementations of workflow systems exist [2,3,4,5]. These systems vary in the formal models by which workflows are described and execution semantics are interpreted, sometimes even allowing multiple execution models within one system [2]. This is in part a result of the different types of applications they try to support, which can have very different requirements. It is also in part due to a lack of standards, each system having its own workflow definition language. The execution models in the Business Process community control easily map onto Petri Nets and thus formal reasoning about workflows is mostly done using Petri Nets [6]. But where business workflows are about describing actual business processes, scientific experiments are more constrained and need to be exactly reproducible. Within e-science not only are there more diverse execution models, there also is a need to use different models within one application. This can be catered for by systems such as Kepler [2]. Furthermore, the need to combine different workflow management systems within one scientific experiment is emerging. Solutions such as the workflow bus [1] are being developed within our research group to suit these needs.

Working with different execution models within one experiment can seriously complicate the design procedure. A way to prove the correctness of these complicated experiments is needed, as well as assistance in exploring the workflow design space. Therefore we need a formal model to reason about the associated

Y. Shi et al. (Eds.): ICCS 2007, Part III, LNCS 4489, pp. 216–219, 2007.

design issues. In previous work we used Turing machines as a formal model to reason about workflow design [7]. One of the advantages to this approach was that we could isolate the execution model in our formal description, allowing us to reason about every possible type of workflow. Being able to reason on this level can give a formal basis for studying meta workflow problems such as the workflow bus [1].

In this paper we introduce an existing formal model called Input Output Automata [8](abbreviated to I/O Automata), to perform the task of reasoning about workflow design.

2 Workflow Design Problem

To reason about workflow design one needs a formal model to represent workflows. One often employed formalism is petri-nets. They are well suited to study control flow in workflows [6]. There are more issues involved in workflow design:

- Connectivity, are two workflow components compatible both in data type and runtime behavior.
- Workflow Provenance, can the experiment be exactly reproduced using the provenance data recorded by a workflow system.
- Workflow representation, what level of detail is desired in a workflow description.

For our research into a workflow bus where multiple (sub) workflows with different execution models are connected, these other issues play a more important part. We propose the use of I/O automata as a formal representation for reasoning about workflow representation and runtime behavior.

I/O Automata were first introduced by Lynch and Tuttle [8], and have been used for the study of concurrent computing problems. They form a labeled state transition system consisting of a set of states, a set of actions divided into input- internal- and output actions(as illustrated in figure3) and a set of transitions which consists of triples of state, action and state. This allows us to study the inherently concurrent nature of workflow systems. One of the characterizing properties of I/O Automata is that input actions are "input enabled", they have to accept and act upon any input. Figure 4 illustrates both that the "input enabled" property defines connections between automata as well as one I/O automaton being computationally equivalent to a composition of several automata.

We study reasoning about representation as a hierarchical problem. Here abstract descriptions should be computationally equivalent to detailed low level descriptions of workflows. This requirement is satisfied by a property called compositionality. In figure 4 compositionality for I/O automata is illustrated.

The composition of a workflow representation starts with a desired input and output as well as a set of available building blocks. The representation of a workflow needs to strike a balance between generality and specificness, resulting in an ideal workflow which is neither to abstract nor to detailed. This idea is illustrated in figure 2, both a top down and bottom up approach are possible.

In the first, the initial input and output requirements are refined into multiple workflow steps. In the second, existing resources are combined (automatically) until they satisfy requirements. The design process can be formalized in a lattice as we did in our previous work [7]. This lattice is set up between the most abstract and most detailed representations of the computational process that a workflow satisfies. In other words, in this design lattice only the representation differs. All different representations within this lattice are computationally equivalent.

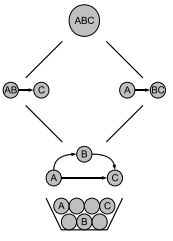

Fig. 1. Workflow design lattice

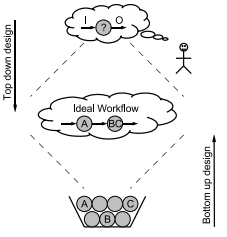

Fig. 2. Workflow design problem

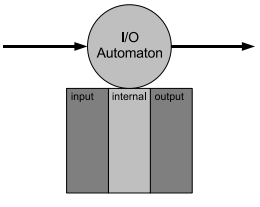

Fig. 3. I/O Automaton

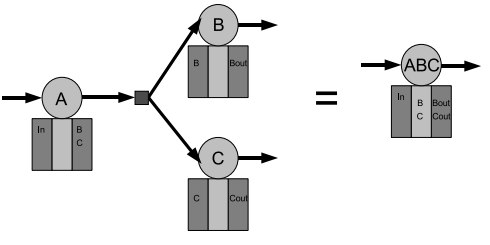

Fig. 4. Illustration of compositionality principle

Workflow components are very general computational elements, which can be modeled as an I/O Automaton. Workflow systems in practice use different execution models based on either data-flow or control flow. To model these execution models I/O Automata representing workflow components need constraints placed on them. In [9] it is shown how the Kahn principle, used as a basis for some data-flow execution models, can be modeled using I/O Automata. The main constraints are that the automata have to be deterministic and all connections are one to one.

3 Conclusions and Future Work

I/O Automata may not provide the answer to all problems involved in creating a workflow bus and other formalisms may be needed. However I/O Automata are suitable for reasoning about workflow representation as well as the runtime

behavior of complicated workflows involving multiple workflow engines and execution models. Using I/O Automata as a formalism, automatic workflow composition can be easily modeled and studied. In future work we plan to give a detailed overview of which formalisms are best suited to what part of workflow design. Furthermore we will show how existing tools for I/O Automata can be used to provide practical support in the workflow design process.

Acknowledgments. This work was carried out in the context of the Virtual Laboratory for e-Science project (www.vl-e.nl). Part of this project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ).

References

1. Zhao, Z., Booms, S., Belloum, A., de Laat, C., Hertzberger, B.: Vle-wfbus: a scientific workflow bus for multi e-science domains. In: E-science 2006, 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam Netherlands. (2006)
2. Ludascher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger-Frank, E., Jones, M., Lee, E., Tao, J., Zhao, Y.: Scientific workflow management and the kepler system. Concurrency and Computation: Practice and Experience, Special Issue on Scientific Workflows **18**(10) (08 25 2006)
3. Majithia, S., Shields, M.S., Taylor, I.J., Wang, I.: Triana: A Graphical Web Service Composition and Execution Toolkit. In: Proceedings of the IEEE International Conference on Web Services (ICWS'04), IEEE Computer Society (2004) 514–524
4. Afsarmanesh, H., Belleman, R., Belloum, A., Benabdelkader, A., van den Brand, J., Eijkel, G., Frenkel, A., Garita, C., Groep, D., Heeren, R., Hendrikse, Z., Hertzberger, L., Kaandorp, J., Kaletas, E., Korkhov, V., de Laat, C., Sloot, P., D.Vasunin, Visser, A., Yakali., H.: Vlam-g: A grid-based virtual laboratory. Scientific Programming (Special issue on Grid Computing) **10** (2002) 173–181
5. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P.: Taverna: A tool for the composition and enactment of bioinformatics workflows. Bioinformatics Journal. **online** (June 16, 2004)
6. Aalst, W.M.P.V.D., Hofstede, A.H.M.T., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distrib. Parallel Databases **14**(1) (2003) 5–51
7. Terpstra, F.P., Adriaans, P.: Designing worklfow components for e-science. In: E-science 2006, 2nd IEEE International Conference on e-Science and Grid Computing, Amsterdam Netherlands. (2006)
8. Lynch, N.A., Tuttle, M.R.: An Introduction to Input/Output Automata. CWI Quarterly **2**(3) (1989) 219–246
9. Lynch, N.A., Stark, E.W.: A proof of the kahn principle for input/output automata. Information and Computation **82**(1) (1989) 81–92