On Characterising and Identifying Mismatches in Scientific Workflows

Khalid Belhajjame, Suzanne M. Embury, and Norman W. Paton

School of Computer Science
University of Manchester
Oxford Road, Manchester, UK
{khalidb, embury, paton}@cs.man.ac.uk

Abstract. Workflows are gaining importance as a means for modelling and enacting *in silico* scientific experiments. A major issue which arises when aggregating a collection of analysis operations within a workflow is the compatibility of their inputs and outputs: the analysis operations are supplied by independently developed web services which are likely to have incompatible inputs and outputs. We use the term mismatch to refer to such incompatibility. This paper characterises the mismatches a scientific workflow may suffer from and specifies mappings for their resolution.

1 Introduction

Scientific workflows are gaining considerable momentum as a mechanism for specifying and automating the execution of scientific experiments [1,10]. During the design of a scientific workflow, the designer's focus is on selecting and composing the analysis operations that will carry out the work of the experiment. Analysis operations are supplied by third parties and as such it is often the case that their inputs and outputs are incompatible with those of the other operations to which they must be connected. We use the term mismatch to refer to such incompatibility. In order to resolve a mismatch, the designer must expend some effort in discovering or implementing special operations that can be plugged into the workflow at the point of incompatibility, and can transform the data sets as necessary to resolve it.

Manual detection and correction of such mismatches is time-consuming and unreliable, and thus reduces the claimed benefits of scientific workflows in facilitating the rapid specification of experiments. In this paper, we propose a classification of the kinds of mismatches that can occur in data-driven workflows and derive the additional information that must be captured about workflow operations if potential mismatches are to be identified automatically. This additional information takes the form of annotations on web service inputs and outputs, based on three separate ontologies.

The remainder of the paper is organised as follows. First, in Section 2, we formally define scientific workflows. In Section 3, we describe the three additional ontologies used for annotating operation inputs and outputs, and use them (in Section 4) to present the mismatch classification and (in Section 5) to specify further annotations for transformation functions that characterise the kinds of mismatches they can address. Finally we close the paper by comparing our work against existing works, and drawing conclusions in Section 6.

U. Leser, F. Naumann, and B. Eckman (Eds.): DILS 2006, LNBI 4075, pp. 240–247, 2006.

2 Scientific Workflows

A scientific workflow is a set of operations connected together using data links. For the purposes of this paper, we define a scientific workflow *SWf* as *SWf* = ⟨*nameWf, OP, DL*⟩, where *nameWf* is a unique identifier for the workflow, *OP* is the set of operations from which the workflow is composed, and *DL* is the set of data links connecting the operations in *OP*.

Operation. An operation $op \in OP$ is defined as $op = \langle nameOp, loc, in, out \rangle$, where *nameOP* is the unique identifier for the operation, *loc* is the URL of the web service that implements the operation, and *in* and *out* are two sets representing the input and output parameters of the operation, respectively.

Parameter. A parameter provides information on the data type of a given operation input/output. It is defined by the pair ⟨*nameP, type*⟩, where *nameP* is the parameter's identifier (unique within the operation) and *type* is the parameter's data type. In our work, we assume an XML type system, so that parameter data types may be either simple types, such as *xs:string* and *xs:int*, or complex types, built from simple ones.

Data Links. Let $IN = \cup_{(op \in OP)} op.in$ be the set of inputs of all the operations comprising a scientific workflow, and $OUT = \cup_{(op \in OP)} op.out$ be the set of outputs of all its operations. The set of data links connecting the workflow operations must then satisfy the following: $DL \subseteq (OP \times OUT) \times (OP \times IN)$. A data link relating the output o of the operation *op1* to the input i of the operation *op2* is therefore denoted by the quadruple *(op1,o,op2,i)*.

3 Ontologies for Characterising Mismatches

Information on the types of operation parameters is usually easily available to scientific workflow systems. For example, where operations are actually web services, the data types can be extracted from the WSDL specification of the service. However, as we have seen, not all mismatches are visible in the types of the connected parameters. In order to automatically detect mismatches, the implicit information about the form and role of operation parameters must be made explicit, just as the information about the type of the parameter currently is. This additional information concerns the semantics of the parameter (i.e. the real world entity to which the parameter corresponds), the representation format used for the parameter over and above any data typing given to it and the extent of the parameter (i.e. the set of possible values which the parameter may take). For each of these, we must create an *ontology* of terms that can be used to annotate services with the information required to detect mismatches.

An ontology is commonly defined as an explicit specification of a conceptualisation [4]. Formally, an ontology θ can be defined as a set of concepts, $\theta = \{c1, \ldots, cn\}$. We use the following ontologies to annotate service parameters for mismatch detection.

Domain Ontology, θ_{domain}. This ontology captures information about the application domains covered by the operations, and enables us to describe the real world concepts to which each parameter corresponds. An example of such an ontology is that developed