Scientific workflow design: theoretical and practical issues
Terpstra, F.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (http://dare.uva.nl)
Summary

This thesis is about practical and theoretical issues in scientific workflow design. In order to explain the importance of scientific workflow design one needs to know the concept of e-Science. This stands for enhanced Science (not electronic as one might expect). It aims to enhance the scientific experiments of scientists from many domains by offering them access to Grid resources. These can enhance experiments through massive computing and massive data handling capabilities. The other enhancement e-Science offers is the sharing of (software) resources that run on the Grid between different scientists. Both for scientists that are collaborating in an experiment but also for scientists from different domains that can re-use each others work. The interface through which scientists from these different domains access Grid resources and share them with their peers is a Scientific Workflow Management System. E-Science is performed by designing and running workflows in such a system. The main research question asked in this thesis is “What is a proper design methodology for both workflow components and workflow topologies that supports the sharing of software resources?” To answer this question we look both at the theoretical and practical sides of scientific workflow design. On the practical side we look into sharing one particular type of software resource, Data assimilation. The idea is that lessons from this specific case can tell us more about the problem of sharing resources in general. First however we turn our attention to the theoretical aspects involved.

The scientific method in general is characterized by the empirical cycle. In this cycle a scientist starts from an existing theory or a research question, from this he derives a hypothesis for testing in an experiment. After performing the experiment a scientist analyzes the result, this can lead to verification or falsification of parts of the existing theory. The theory is adjusted accordingly and the whole cycle can start again performed either by the same scientist or a peer performing a peer review of the results of the experiment. In e-Science designing a workflow is analogous to formulating a hypothesis, just as performing a workflow equates to performing an experiment in general science. There also are differences between science in general and e-Science, in particular concerning the sharing of (software) resources. The main method of sharing in science in general is through peer
reviewed scientific papers. Within e-Science the work of scientists can also be shared directly as resources ready to be used in a workflow. Widespread adoption of this type of sharing can lead to a paradigm shift for science as a whole, by allowing a far greater search space to be covered in scientific experiments. The start of this can be seen in bio-informatics where the sharing of research through webservises is leading to many new discoveries. Currently the sharing of resources has not yet risen to significant levels for science as a whole. While in bio-informatics mainly data is shared, sharing of methods and models as needed for sharing resources related to data assimilation, is more complicated. Currently e-Science can not convincingly be called a paradigm shift however if sharing of more complex resources becomes widespread this could change in the future.

In order to do workflow design one needs to know what the workflow design space looks like. In this thesis an overview is given of the different approaches to workflow design, concrete, abstract to concrete and automatic workflow composition. Using a formal approach we show that automatic workflow composition in the general case is not possible. Furthermore limits of abstraction in workflow representation are proven. Maximal generality and simplicity are mutually exclusive with consistency. In other words the most general and abstract representation of a workflow can not be consistent. From this formal look at the workflow design space a number of criteria are derived that need to be met by a formalism for reasoning about specific workflow instances. As was mentioned earlier workflow design is analogous to formulating a hypothesis. A scientist needs to formulate a hypothesis in such a way that experiments based on this hypothesis will provide answers to the research question. The execution of workflow needs to confirm or falsify existing theories. Formal reasoning about a workflow can help a scientist determine whether his workflow is consistent and actually provides relevant answers especially in more complicated workflows.

To this end five formalisms are compared for use in abstract to concrete workflow design, the method that is most suitable to scientific workflows, especially when sharing resources. First Petri nets, a formalism used extensively in concrete workflow design, especially in the business process workflow field. Second $\pi$ Calculus a more recent alternative to Petri nets used for the same applications. Third there are Turing machines which provided the underpinnings of the proofs in the formal analysis of the workflow design space. The final two I/O Automata and Constraint Automata have not been applied to workflows yet but have many of the properties that are needed. From this comparison Constraint Automata emerge as the most suitable formalism for the abstract to concrete approach to workflow design.

From the theoretical side of workflows and workflow design we move to the practical side. An analysis is given of the types of support workflow systems can give to a scientist that is designing a workflow. This support can be during composition of a workflow, the development of resources, the
execution of a workflow as well as the dissemination of a workflow. Current workflow systems are compared on the basis of the features they provide in supporting a scientist when creating a workflow.

For the practical aspects of sharing one particular software resource, we look at the technique of data assimilation. This is done through the use of two case studies. Before going into the details of the case studies the background to data assimilation is explained. How it originated in weather prediction as a way to minimize errors in both the predictive model as well as in the observations used for prediction. The elements which make up data assimilation are explained, the predictive model as well as the estimator which can adjust parameters of this model as well as correct errors in observation. An overview of current data assimilation toolkits is given. Within this overview features, relevant for employing them as a shared resource in an e-Science environment, are compared. Two of these toolkits are used in the case studies.

The first case study is about bird migration. In this study data assimilation is used to minimize observation error. The aim is to predict bird densities of migratory birds over the Netherlands using a model for bird migration from Scandinavia to Africa. The second case study looks at the prediction of car traffic for one particular stretch of Dutch highway. The aim is to predict traffic several hours ahead. In this study data assimilation is used to minimize error in the employed predictive model. From these case studies we learn what types of knowledge about implementing data assimilation should be made explicit in order to use data assimilation as a shared resource.

This knowledge is combined with the earlier lessons learned about the workflow designspace and the use of formalisms in an abstract to concrete design methodology to produce a methodology for implementing data assimilation as a shared resource. It is a workflow detailing how to implement each step in the data assimilation process in a scientific workflow. Data preparation model development choice of estimator, use of parallelism are all detailed.

The thesis is concluded by looking at what lessons can be drawn from the specific case of data assimilation as well as from the exploration of the workflow designspace to answer the original research question: what is a proper methodology to share software resources in the general case. Not only are proper methodologies needed but also changes in the way science is funded. Such a change can ensure there is more incentive for scientists to share resources. Funding is not the only measure that can provide incentive, a change in the scientific review process and the role of publishers can also help. Not only scientific papers need to be reviewed, but also shared resources as well as workflows used that were employed in scientific discoveries. Publishers of scientific journals can offer more than just scientific papers, they can also publish peer reviewed resources and workflows.