SCIENTIFIC WORKFLOWS

Bertram Ludäscher
Dept. of Computer Science, UC Davis
UC Davis Genome Center

Shawn Bowers
UC Davis Genome Center

Timothy McPhillips
UC Davis Genome Center

SYNONYMS

Related terms: *in silico experiment*, *grid workflow*

DEFINITION

A *scientific workflow* is the description of a process for accomplishing a scientific objective, usually expressed in terms of *tasks* and their *dependencies*. Typically, scientific workflow tasks are computational steps for scientific simulations or data analysis steps. Common elements or stages in scientific workflows are acquisition, integration, reduction, visualization, and publication (e.g., in a shared database) of scientific data. The tasks of a scientific workflow are organized (at design time) and orchestrated (at runtime) according to dataflow and possibly other dependencies as specified by the workflow designer. Workflows can be designed visually, e.g., using block diagrams, or textually using a domain-specific language.

HISTORICAL BACKGROUND

Workflows have a long history in the database community and in business process modeling, in which case they are sometimes called *business workflows* to distinguish them from scientific workflows. The database community realized early [10] that scientific data management has different characteristics from more traditional business data management. Early work on scientific workflows within the database community took a database-centric view by defining data models and query languages suitable for scientific experiment management systems: the MOOSE data model and FOX query language have their roots in the late eighties [5] and early nineties [13] and gave rise to the ZOO experiment management environment [6], an early system based on an underlying object-oriented database. Another pioneering work that emphasized the importance of workflow concepts in scientific data management is WASA, a ***W****orkflow-based* ***A****rchitecture for* ***S****cientific* ***A****pplications* [8]; the related publication [12] introduced the term 'scientific workflow' and contrasted such workflows with office automation and business workflows. An early benchmark comparing different database architectures for scientific workflow applications is LabFlow-1 [1].

Other roots of scientific workflow systems include *problem solving environments*, which emerged in the nineties in the computational sciences community as intuitive tools to “solve a target class of problems for scientific computing” [4], and *laboratory information management systems* (LIMS) [9], which can be seen as special scientific workflow systems that are used in a laboratory environment for the management of samples, instrument-based measurements, and other functions, including data analysis and workflow automation. Similar to many scientific workflow systems, problem solving environments and LIMS sometimes employ a visual programming paradigm to link together components. An early, if not the first, visual language that allowed simple interfacing with lab instruments was G in LabVIEW1.0, released in 1986 for the Apple Macintosh. Modern incarnations of LIMS can include functions of enterprise resource planning (ERP) systems and thus go beyond the scope of current scientific workflow systems.

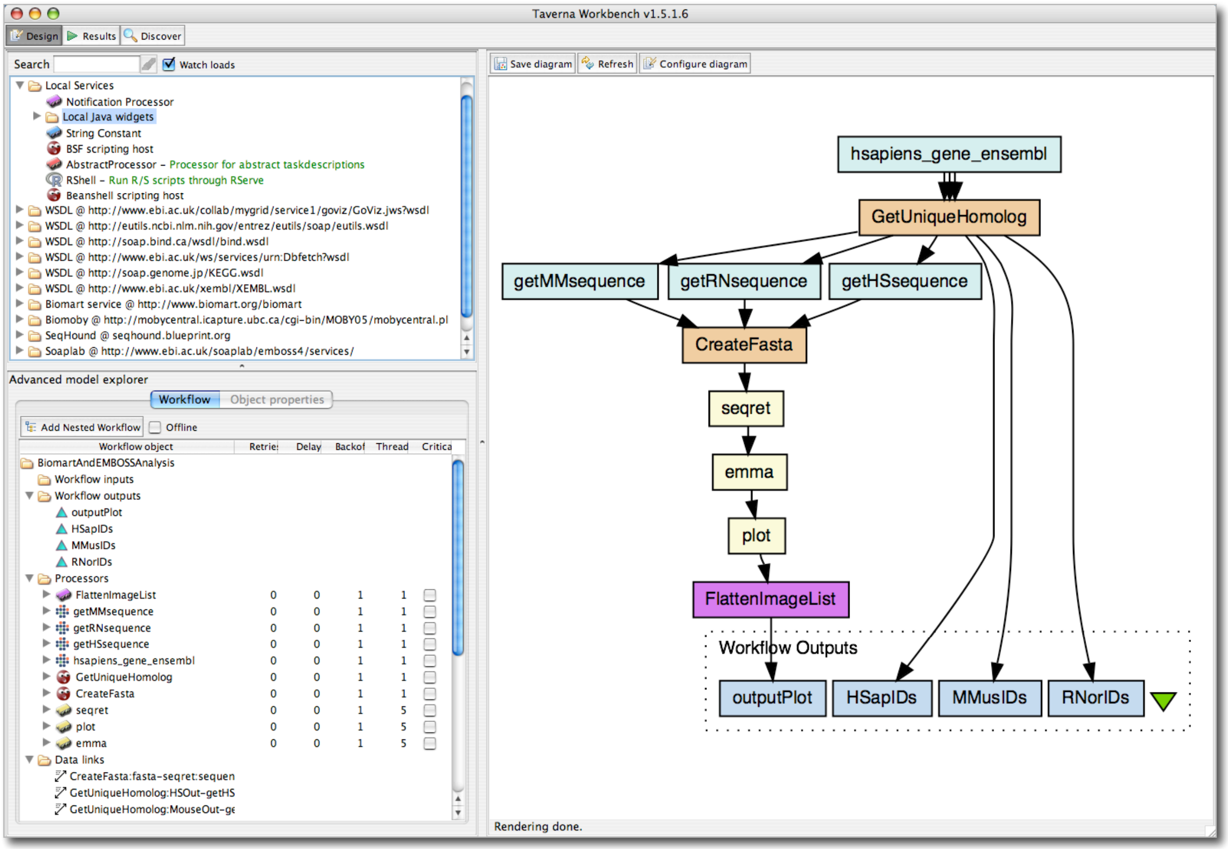
With the advent of *e-Science* as a paradigm, scientific workflow research and development has seen a major resurgence. Similar to the related term *cyberinfrastructure*, e-Science brings together computational techniques and tools from the computational sciences, distributed and high-performance computing, databases, data analysis, visualization, sharing, and collaboration. There are now a number of new open source as well as commercial scientific workflow systems available and under active development. For example, a special journal issue of *Concurrency and Computation: Practice and Experience* covers a number of systems, including Kepler, Taverna, and Triana among others [2]. For a high-level overview and attempt at a classification of current scientific workflow systems see [14], which includes also references to many other systems, such as Askalon, Pegasus/DAGMan, Karajan, etc.

SCIENTIFIC FUNDAMENTALS

Science is an exploratory process involving cycles of observation, hypothesis formation, experiment design and execution. Today, scientific knowledge discovery is increasingly driven by data analysis and computational methods, e.g., due to ever more powerful instruments for observation and the use of commodity clusters for high-performance scientific computing and simulations in the computational sciences. Scientific workflows can be applied during various phases of the larger science process, specifically modeling and automation of computational experiments, data analysis, and data management. The results from workflow runs can yield new data and insights and thus may lead to affirmation, modification, or refutation of a given hypothesis or experiment outcome.

Scientific workflow systems automate the execution of scientific workflows, and may additionally assist in workflow design, composition, and the management and sharing of workflow descriptions. Other important functions include support for workflow execution monitoring, for recording and querying provenance information, for workflow optimization (e.g. exploiting dataflow and concurrency information for parallel execution), and for fault-tolerant execution. These additional features also distinguish a scientific workflow systems approach from more traditional script-based solutions in which such functionality is usually not provided. Workflow provenance information can be used, e.g., to facilitate the interpretation, debugging, and reproducibility of scientific analyses. An increasing number of scientific workflow systems now offer support for various forms of provenance. One can distinguish *data provenance*, i.e., the processing history of data, and provenance information describing the *workflow evolution*, i.e., the history of changes of a workflow definition and the parameter settings used for a particular workflow instance.

Scientific workflows are often visually represented as directed graphs (Figures 1 and 2) linking



Local Services
- Notification Processor
- Local Java widgets
- String Constant
- BSF scripting host
- AbstractProcessor – Processor for abstract taskdescriptions
- RShell – Run R/S scripts through RServe
- Beanshell scripting host

WSDL @ http://www.ebi.ac.uk/collab/mygrid/service1/goviz/GoViz.jws?wsdl
WSDL @ http://eutils.ncbi.nlm.nih.gov/entrez/eutils/soap/eutils.wsdl
WSDL @ http://soap.bind.ca/wsdl/bind.wsdl
WSDL @ http://www.ebi.ac.uk/ws/services/urn:Dbfetch?wsdl
WSDL @ http://soap.genome.jp/KEGG.wsdl
WSDL @ http://www.ebi.ac.uk/xembl/XEMBL.wsdl
Biomart service @ http://www.biomart.org/biomart
Biomoby @ http://mobycentral.icapture.ubc.ca/cgi-bin/MOBY05/mobycentral.pl
SeqHound @ seqhound.blueprint.org
Soaplab @ http://www.ebi.ac.uk/soaplab/emboss4/services/

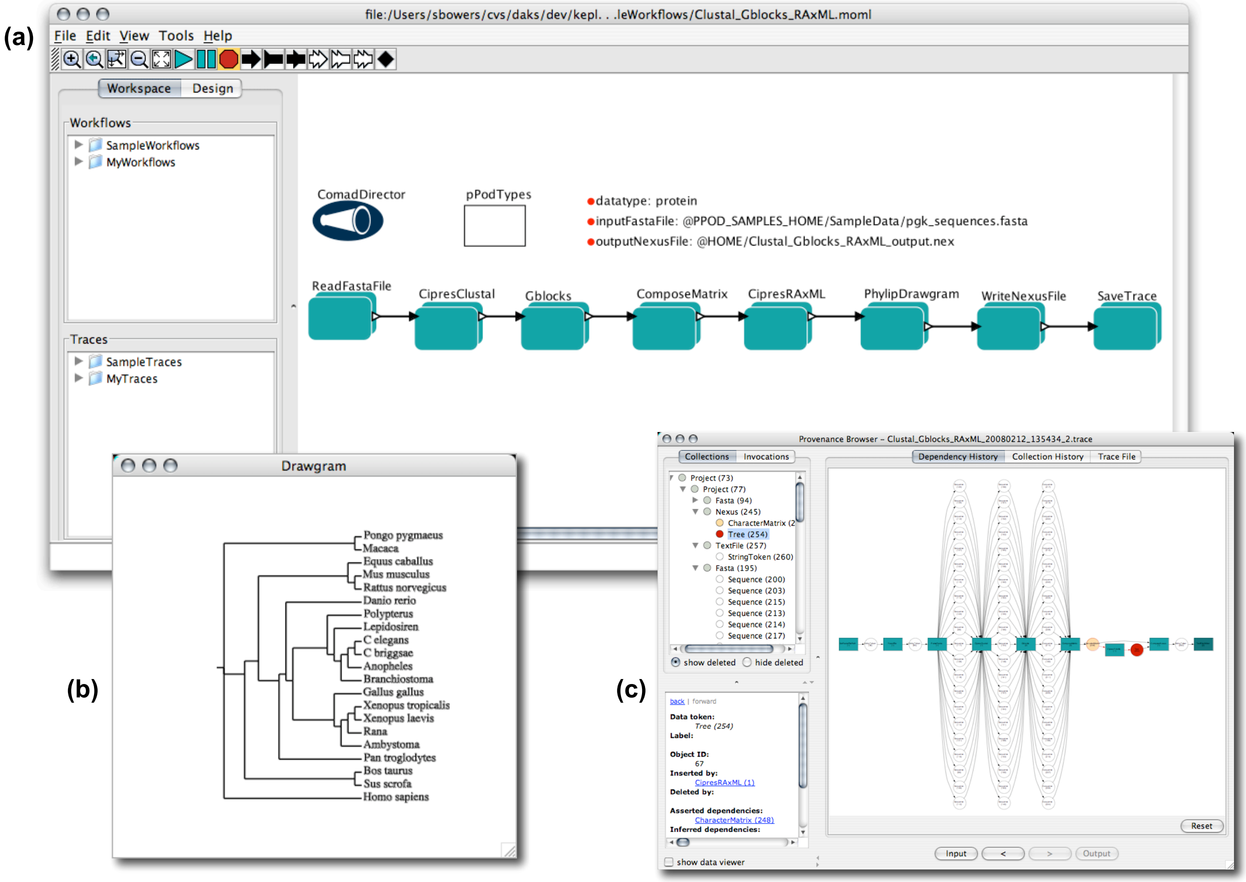
Workflow object	Retrie:	Delay	Backof	Thread	Critica
BiomartAndEMBOSSAnalysis					
Workflow inputs					
Workflow outputs					
outputPlot					
HSapIDs					
MMusIDs					
RNorIDs					
Processors					
FlattenImageList	0	0	1	1	□
getMMsequence	0	0	1	1	□
getRNsequence	0	0	1	1	□
getHSsequence	0	0	1	1	□
hsapiens_gene_ensembl	0	0	1	1	□
GetUniqueHomolog	0	0	1	1	□
CreateFasta	0	0	1	1	□
seqret	0	0	1	5	□
plot	0	0	1	5	□
emma	0	0	1	5	□
Data links					
CreateFasta:fasta-seqret:sequen					
GetUniqueHomolog:HSOut-getHS					
GetUniqueHomolog:MouseOut-ge					

Rendering done.

Figure 1: Example workflow represented in the Taverna workflow system. This workflow extracts gene IDs from human chromosome 22 with mappings to disease functions and homologues in mouse and rat; fetches base pairs of the associated DNA sequences; combines the sequences into a FASTA file; performs a multiple sequence alignment; and renders the result. The workflow uses three soaplab-based analysis operations (seqret, emma, plot) that run on the EBI compute cluster.

atomic tasks or composite components, so-called *subworkflows*. Tasks can include native functions of the workflow system, but often correspond to invocations of local applications, remote (web) services, or subworkflows. Scientific workflows differ from conventional programming in that the workflows are often more coarse-grained and involve wiring together of pre-existing components and specialized algorithms. Figure 1 shows a simple bioinformatics workflow in the Taverna system, consisting of multiple (soaplab) services.

There is currently no standard *scientific workflow language*, and standards from related communities (e.g., BPEL4WS) have not found widespread adoption in the scientific workflow community. For example, job-based *grid workflows* are often represented as directed acyclic graphs (DAGs), which are then scheduled on a computational grid or cluster computer according to the implied task dependencies. In this *model of computation*, each task is executed only once per workflow run and task scheduling amounts to finding a *topological sort* for the partial order implied by the DAG. Other, more sophisticated models of computation consider tasks as independent and continuously executing processes which can receive and send many different data items per workflow run. Scien-



datatype: protein
inputFastaFile: @PPOD_SAMPLES_HOME/SampleData/pgk_sequences.fasta
outputNexusFile: @HOME/Clustal_Gblocks_RAxML_output.nex

◉ show deleted ○ hide deleted

Data token:
Tree (254)
Label:

Object ID:
67
Inserted by:
CipresRAxML (1)
Deleted by:

Asserted dependencies:
CharacterMatrix (248)
Inferred dependencies:

☐ show data viewer

Figure 2: Example scientific workflow in the Kepler system: (a) user interface for creating, editing, and executing scientific workflows; (b) a visual representation of the data product (a phylogenetic tree) computed by a workflow run; and (c) a viewer for navigating the data provenance (lineage) captured in an execution trace. This workflow uses a combination of local and remote (web) services to perform multiple sequence alignment and phylogenetic tree inference on input DNA sequences.

tific workflow systems that support such models of computation may thus be used for *data stream processing* and *continuous queries*. Similar to business workflows, formal approaches such as *Petri nets* can be used to describe scientific workflow execution semantics. However, the dataflow models of computation of many scientific workflow systems can exhibit both task- and pipeline-parallelism where token order is important. A standard computation model for such dataflow systems is the *Kahn Process Network* model. The structurally simple linear Kepler workflow in Figure 2 is achieved via a special model of execution, implemented by a so-called *director*.[1] The COMAD (*Collection-Oriented Modeling And Design*) director in Figure 2 specifies that workflow components work on a continuous, XML-like data stream which passes through all components eventually. Each component is configurable to compute only on certain (tagged) data collections; results are injected back into the stream. The resulting more linear workflows are easy to comprehend and evolve over

[1] Kepler inherits from the underlying Ptolemy II system the capability to use distinct directors at different workflow modeling levels and thus to combine different models of computation in a single workflow.

time, another important advantage over script-based solutions.

KEY APPLICATIONS

Scientific workflows now span virtually all areas of the natural sciences. Bioinformatics is a particularly active application area (cf. Figures 1 and 2), but the spectrum of disciplines employing scientific workflow systems is much wider and includes particle physics, chemistry, neurosciences, ecology, geosciences, oceanography, atmospheric sciences, astronomy and cosmology, among others.

URL to CODE

A number of open source scientific workflow systems are available, among them:
Kepler: `http://www.kepler-project.org`
Taverna: `http://taverna.sourceforge.net`
Triana: `http://www.trianacode.org`
For a list including many other systems, see `http://www.extreme.indiana.edu/swf-survey/`.

CROSS REFERENCES

Business Process Modeling
(Business) Workflow
Data Analysis
Dataflow
Problem Solving Environment
Provenance
Visual Programming

RECOMMENDED READING

A collection of articles on scientific workflow applications, formal foundations, and scientific workflow systems can be found in a recent book [11]. The findings of an NSF-funded workshop on scientific workflows are reported in [3]. Other special issues on scientific workflows appeared, e.g., in *Concurrency and Computation: Practice and Experience* [2] and in the *ACM SIGMOD Record* [7].

References

[1] A. J. Bonner, A. Shrufi, and S. Rozen. LabFlow-1: A Database Benchmark for High-Throughput Workflow Management. In *5th Intl. Conf. on Extending Database Technology (EDBT), Avignon, France*, pages 463–478. Springer LNCS 1057, 1996.

[2] G. C. Fox and D. Gannon, editors. *Concurrency and Computation: Practice and Experience. Special Issue: Workflow in Grid Systems*, volume 18(10). John Wiley & Sons, 2006.

[3] Y. Gil, W. Deelman, W. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers. Examining the Challenges of Scientific Workflows. *Computer*, 40(12):24–32, 2007.

[4] E. Houstis, E. Gallopoulos, R. Bramley, and J. Rice. Problem-solving Environments For Computational Science. *IEEE Computational Science & Engineering*, 4(3):18–21, 1997.

[5] Y. E. Ioannidis and M. Livny. MOOSE: Modeling Objects in a Simulation Environment. In G. X. Ritter, editor, *IFIP Congress*, pages 821–826. North Holland, August 1989.

[6] Y. E. Ioannidis, M. Livny, S. Gupta, and N. Ponnekanti. ZOO: A Desktop Experiment Management Environment. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *Proceedings of International Conference on Very Large Data Bases (VLDB)*, pages 274–285, 1996.

[7] B. Ludäscher and C. Goble, editors. *ACM SIGMOD Record: Special Issue on Scientific Workflows*, volume 34(3), September 2005.

[8] C. B. Medeiros, G. Vossen, and M. Weske. WASA: A Workflow-Based Architecture to Support Scientific Database Applications. In *Database and Expert Systems Application (DEXA)*, pages 574–583. Springer LNCS 978, 1995.

[9] A. S. Nakagawa. *LIMS: Implementation and Management.* The Royal Society of Chemistry, Thomas Graham House, The Science Park, Cambridge CB4 4WF, 1994.

[10] A. Shoshani, F. Olken, and H. K. T. Wong. Characteristics of Scientific Databases. In *10th Intl. Conf. on Very Large Data Bases (VLDB)*, pages 147–160. Morgan Kaufmann, 1984.

[11] I. Taylor, E. Deelman, D. Gannon, and M. Shields, editors. *Workflows for e-Science: Scientific Workflows for Grids.* Springer, 2007.

[12] J. Wainer, M. Weske, G. Vossen, and C. B. Medeiros. Scientific Workflow Systems. In *Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems: State of the Art and Future Directions*, 1996.

[13] J. L. Wiener and Y. E. Ioannidis. A Moose and a Fox Can Aid Scientists with Data Management Problems. In C. Beeri, A. Ohori, and D. Shasha, editors, *4th Intl. Workshop Database Programming Languages (DBPL)*, pages 376–398. Springer, 1993.

[14] J. Yu and R. Buyya. A Taxonomy of Scientific Workflow Systems for Grid Computing. In Ludäscher and Goble [7].