

## Taverna / myGrid: aligning a workflow system with the life sciences community

Tom Oinn<sup>1</sup>, Peter Li<sup>2</sup>, Douglas B. Kell<sup>2</sup>, Carole Goble<sup>3</sup>, Antoon Goderis<sup>3</sup>, Mark Greenwood<sup>3</sup>, Duncan Hull<sup>3</sup>, Robert Stevens<sup>3</sup>, Daniele Turi<sup>3</sup> and Jun Zhao<sup>3</sup>

<sup>1</sup> EMBL European Bioinformatics Institute, Hinxton, Cambridge, CB10 1SD, UK  
tmo@ebi.ac.uk

<sup>2</sup> Bioanalytical Sciences, School of Chemistry, University of Manchester, M13 9PL, UK  
Peter.Li@manchester.ac.uk

<sup>3</sup> School of Computer Science, University of Manchester, M13 9PL, UK  
carole@cs.man.ac.uk

### 20.1 Introduction

Bioinformatics is a discipline that uses computational and mathematical techniques to store, manage and analyse biological data in order to answer biological questions. Bioinformatics has over 850 databases [181] and numerous tools that work over those databases and local data to themselves produce even more data. In order to perform an analysis, a bioinformatician uses one or more of these resources to gather, filter and transform data to answer a question. Thus, bioinformatics is an *in silico* science.

The traditional bioinformatics technique of cutting and pasting between Web pages can be effective, but it is neither scalable nor does it support scientific best practice, such as record keeping. In addition, as such methods are scaled up, slips and omissions are more likely to occur. A final human factor is the tedium of such repetitive tasks [371].

Doing these tasks programmatically is an obvious solution, especially for the repetitive nature of the tasks. Some bioinformaticians have the programming skills to wrap these distributed resources. Such solutions are, however, not easy to disseminate, adapt and verify. Moreover, one of the consequences of the autonomy of bioinformatics service providers is massive heterogeneity within those resources. The advent of Web Services has brought about a major change in the availability of bioinformatics resources from Web pages and command line programmes to Web services [369], though much of the structural, value based and syntactic heterogeneity remains. The consequent lack of a common type system means that services are difficult to join together programmatically and any technical solution to *in silico* experiments in biology has to address this issue.

Many scientific computing projects within the academic community have turned to workflows as a means of orchestrating complex tasks (*in silico* experiments) over a distributed set of resources. Examples include DiscoveryNet [352] for molecular biology and environmental data analysis, SEEK for ecology [67, 68], GriPhyn for particle physics [144], and SCEC/IT for earthquake analysis and prediction [242].

Workflows offer a high-level alternative for encoding bioinformatics *in silico* experiments. The high-level nature of the encoding means a broader community can create templates for *in silico* experiments. They are also easier to adapt or re-purpose by substitution or extension. Finally workflows are less of a black-box than a script or traditional programme; the experimental protocol captured in the workflow is displayed in such a way that a user can see the components, their order and input & outputs. Such a workflow can be seen in Figure 20.1.

<sup>my</sup>Grid is a project to build middleware to support workflow-based *in silico* experiments in biology. Funded by the UK's e-Science Programme from 2001, it has developed a set of open source components that can be used independently and together. These include a service directory [268], ontology-driven search tools over semantic descriptions of external resources and data [268]; data repositories and semantically-driven metadata stores for recording the provenance of a workflow and the experimental lifecycle [457], as well as other components such as distributed query processing [63], event notification<sup>1</sup>.

<sup>my</sup>Grid's workflow execution and development environment, Taverna, links together and executes external remote or local, private or public, third party or home-grown, heterogeneous open services, (applications, databases, etc). The Freefluo workflow enactment engine<sup>2</sup> enacts the workflows. The Taverna workbench is a GUI-based application for bioinformaticians to assemble, adapt and run workflows, and manage the generated data and metadata. <sup>my</sup>Grid components are Taverna plug-ins (for results collection and browsing, provenance capture, service publication & discovery) and services (such as specialist text mining). Thus the workbench is the user facing application for the <sup>my</sup>Grid middleware services. At the time of writing Taverna 1.3 has been downloaded over 14 000<sup>3</sup> times and has an estimated user base of around 1 500 installations. Taverna has been used in many different areas of research throughout Europe and the U.S.A. for functional genomics, systems biology, protein structure analysis, image processing, chemoinformatics and simulation co-ordination. From 2006, <sup>my</sup>Grid has been incorporated into the UK's Open Middleware Infrastructure Institute to be "hardened" and developed to continue to support Life Scientists.

---

<sup>1</sup> <http://www.mygrid.org.uk>

<sup>2</sup> <http://freefluo.sourceforge.net>

<sup>3</sup> see <http://taverna.sourceforge.net/index.php?doc=stats.php>

### 20.1.1 A bioinformatics Case Study

An exemplar Taverna workflow currently being used for systems biology is shown in Figure 20.1. This workflow uses data stored in distributed databases to automate the reconstruction of biological pathways which represent the relationships between biological entities such as genes, proteins and metabolites.

The interaction pathways generated by the workflow are in the form of a data model, which is specified by the XML-based Systems Biology Markup Language [216]. A core SBML workflow is responsible for generating a SBML model. This is then populated, through the SBML API, by the supplementary workflows that gather data for the model (see Figure 20.1). The SBML model can then be used to perform biological simulations.

These workflows typify the needs of bioinformatics analyses. It is a typically data centric workflow gathering many kinds of data from a variety of locations and from services of a variety of technology types. As will be seen through the chapter, many types of resources are used and these can all be incorporated into Taverna. The workflows have to be run repeatedly and such an analysis would be long and tedious to perform manually.

The rest of this chapter is organised as follows. Section 20.2 further elaborates on the background to Taverna, then outlining requirements in detail in Section 20.3. Section 20.4 introduces the major Taverna components and architecture. Section 20.5 concentrates on the workflow design, and Section 20.6 on executing and monitoring workflows. Section 20.7 completes the workflow lifecycle with metadata and provenance associated with managing and sharing results, and the workflows themselves. Section 20.8 discusses related work. Section 20.9 reflects on our experiences and showcases future developments in Taverna 2.0.

## 20.2 The Bioinformatics Background

Life scientists are accustomed to making use of a wide variety of web-based resources. However, building applications that integrate resources with interfaces designed for humans are difficult and error-prone [369]. The emergence of Web Services [98], along with the availability of suitable tool support, has seen a significant number of bioinformatics web resources becoming publicly available, and described with a Web Services Description Language (WSDL) interface.

There are currently over 3000 services accessible to a <sup>my</sup>Grid user. Although the majority involve complex interaction patterns, specific messaging formats, or use different protocols and paradigms, they actually follow a small number of stereotyped patterns. The user's lack of middleware knowledge means they should not be expected to deal with the differences between these patterns. In addition, given the number and distribution of services users

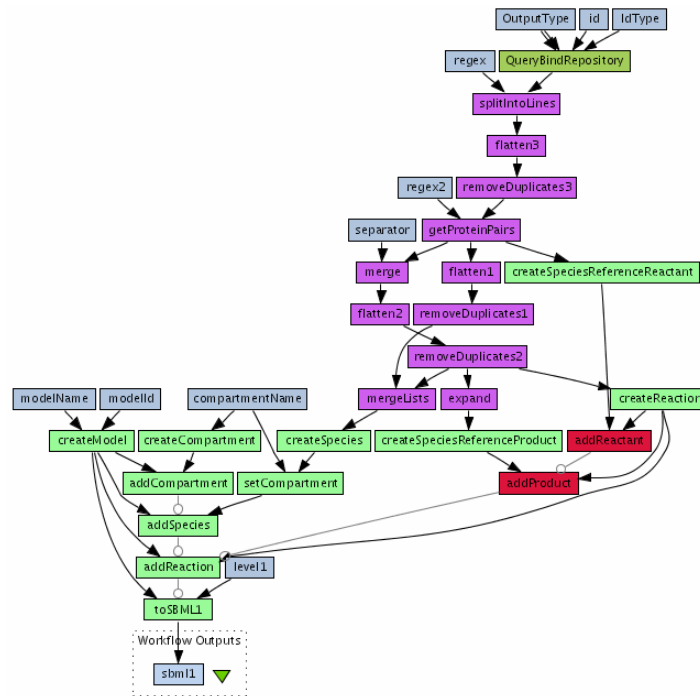


Fig. 20.1: An SBML model construction workflow. This workflow retrieves protein interactions from the BIND database which are then used to populate a SBML model using the core SBML workflow. Four types of processors are used in this example: WSDL (dark green), consumer API (light green), local Java (purple) and nested workflow (red) processors. These processors are joined together by data links (black) and coordination links (grey). Local processors such as `getProteinPairs` and `merge` act as shim services by adapting data for population of the SBML model.

cannot be expected to have existing knowledge of what services are available, where they are or what they do.

The data produced by these services is mostly semi-structured and heterogeneous. There are a large number of data formats including those for gene sequences, for protein sequences, as well as bespoke formats produced by many analysis tools. These are rarely encoded in XML and there is usually no formal specification that describes these formats. Interpreting or reconciling these data as it is passed between different databases and analysis tools is therefore difficult.

This situation is in contrast with data in other scientific workflow projects which have much more centralised control of data formats. For example, the SEEK project provides tools for ecologists within the project to describe their

data using XML Schema and ontologies, and so support middleware driven data integration [99].

DiscoveryNet [352] requires each application service to be wrapped allowing data to adhere to a common format. Other projects are more uniform than myGrid in the way applications on distributed resources are accessed. For example, abstract Pegasus workflows used in the SCEC/IT project are first compiled into concrete workflows. Each step of a concrete workflow corresponding to a job to be scheduled on a Condor cluster [145].

Taverna differs from these projects by placing an emphasis on coping with an environment of autonomous service providers and a corresponding ‘open world’ model for the underlying Grid and Service-Orientated Architecture. Taverna’s target audience of life scientists want easy access and composition of as wide a range of services as feasible and this reinforces the need for an open access policy for services, despite the obvious difficulties.

### 20.3 Aligning with Life Science

From the background and introduction we can define the key requirements for the Taverna workflow system that drive us to align with Life Science:

Ease of Use: The target end user for Taverna are not necessarily expert programmers.

Data Flow Centric: Bioinformaticians are familiar with the notion of data flow centric analysis. We want to enhance how biologists perform their models of analysis, not to change their model of analysis.

Open World Assumption: We want to be able to use any service as presented, rather than require service providers to implement services in a prescribed manner and thus create a barrier to adoption.

Easy and Rapid User-driven *ad-hoc* Workflow Design: Quickly and easily finding services and adapting previous workflows is key to effective workflow prototyping.

Fault Tolerant: Any software operating in a networked, distributed environment is required to cope gracefully with failure.

Support for the e-Science life cycle: Workflows are not a complete solution for supporting *in silico* experiments. They exist in a wider context of scientific data management, as illustrated in Figure 20.2. It is essential that data produced by a workflow carries with it some record of how and why it was produced *i.e.*, the provenance of the data.

### 20.4 Architecture of Taverna

The requirements described have led to several major design lessons. Figure 20.3 illustrates how Taverna takes a layered approach to its overall architecture. This is driven by the need to present a useful, high-level presentation

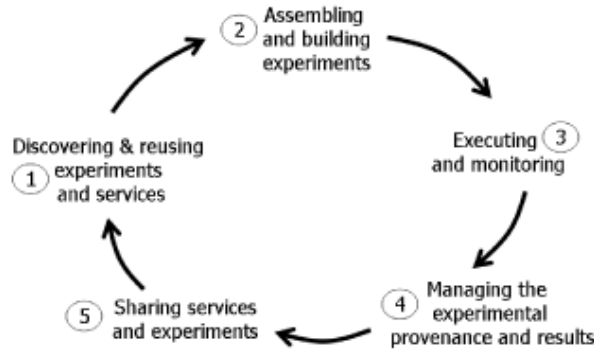


Fig. 20.2: The e-Science life cycle

in which biologists can co-ordinate a variety of resources. Our user-base neither knows nor cares about such things as port types *etc.* We have a requirement to both present a straight forward perspective to our users and yet cope with the heterogeneous interfaces of our services. A major consequence of this for the workflow system architecture has been to provide a multi-tiered approach to resource discovery and execution that separates application and user concerns from operational and middleware concerns.

Scuff, a workflow language for linking applications [314], is at the abstraction level of the user; an extensible processor plug-in architecture for the Freefluo enactor manages the low-level “plumbing” invocation complexity of different families of services. In between lies an execution layer interpreting the Taverna Data Object Model that handles user-implied control flows such as implicit iteration over lists, and a user’s fault tolerance policies.

Figure 20.3 shows how the <sup>my</sup>Grid components are divided between the three layers of <sup>my</sup>Grid’s design.

- The Application Data Flow layer, is aimed at the user and is characterised by a User-Level Workflow Object Model. The purpose is to present the workflows from a problem-oriented view, hiding the complexity of the interoperation of the services. When combining services into workflows, users think in terms of (see figure 20.4) the data consumed and produced by logical services and connecting them together. They are not interested in the implementation styles of the services.
- The Execution Flow layer relieves the user of most of the details of the execution flow of the workflow and expands on control flow assumptions that tend to be made by users. This layer is characterised by the Enactor Internal Object Model and by the <sup>my</sup>Grid Contextual Information Model. The layer manages list and tree data structures, implicitly iterates over collections of inputs and implements fault recovery strategies on behalf

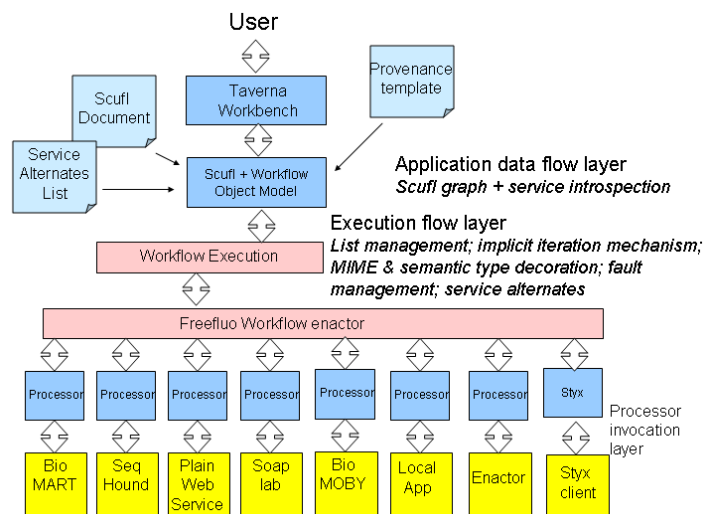


Fig. 20.3: An overview of Taverna in layers.

of the user. This saves the user explicitly handling these at the Application layer and avoids mixing the mechanics of the workflow with its conceptual purpose. A drawback is that an expert bioinformatician needs to understand the behavioural semantics of this layer to avoid duplicating the implicit behaviour.

- The Processor Invocation layer, is aimed at interacting with and invoking concrete services. Bioinformatics services developed by autonomous groups can be implemented in a variety of different styles even when they are similar logical services from a scientist's perspective. This layer is characterised by the Enactor Internal Object Model and is catered for by an extensible processor plug-in architecture for the Freefluo enactment engine.

<sup>my</sup>Grid is designed to have a framework that provides three levels of extensibility:

- The first level provides a plug-in framework to add new GUI panels to facilitate user interaction for deriving and managing the behavioural extensions incorporated into Taverna. This extensibility is made available at the workbench layer.
- The second level allows for new processor types to be plugged-in to enable the enactment engine to recognise and invoke new types of services (which can be both local and external services). This permits a wider variety of workflows to be constructed and executed. This level of extensibility is provided at the workflow execution layer.

- The third level is provided for loosely integrating external components *via* an event-observer interface. The workflow enactor generates events during critical state changes as it executes the workflow, exposing snap shots of important parts of its internal state *via* event objects (*i.e.* messages). Those event objects are then intercepted and processed by observer plugins that can interact with external services. This level of extensibility is made available at the workflow execution layer.

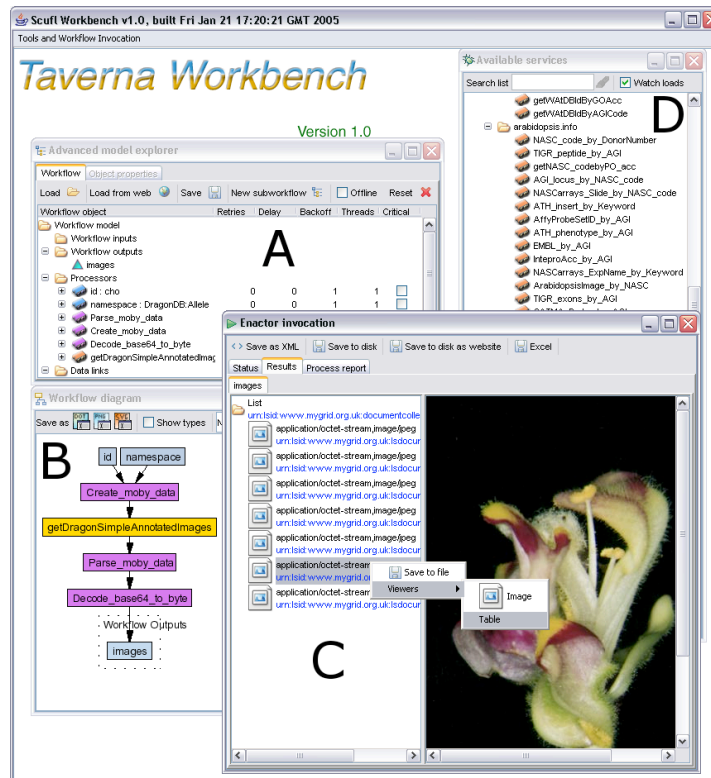


Fig. 20.4: The Taverna Workbench showing a tree structure explorer (A) and a graphical diagram view (B) of a Scuff workflow. The results of this workflow are shown in the enactor invocation window in the foreground (C). A service palette showing the range of operations which can be used in the composition of a workflow is also shown (D).

The Scuff language [314] is essentially a data flow centric language, defining a graph of data interactions between different services (or, more strictly, processors. Scuff is designed to reflect the users abstraction of the *in silico*



experiment, rather than the low-level details of the enactment of that experiment.

Internally to Taverna, Scuff is represented using a Workflow Object Model, along with additional information gained from introspecting over the services. A typical workflow developed in the systems biology use case is shown in Figure 20.1.

The components of a Scuff workflow are:

- A set of inputs that are entry points for the data for the workflow.
- A set of outputs that are exit points for the data for the workflow.
- A set of processors each of which represents a logical service: an individual step within a workflow. A processor includes a set of input ports and a set of output ports. From the user's perspective the behaviour of a processor is to receive data on its input ports, (process the data internally) and to produce data on its output ports.
- A set of data links that link data sources to data destinations. The data sources can be inputs or processor output ports, and data destinations can be outputs or processor input ports.
- A set of coordination links that enable running order dependencies to be expressed where direct data flow is not required by providing additional constraints on the behaviour of the linked processors. For example, in Figure 20.1 two coordination links are defined so that that one processor will not process its data until another processor completes, even though there is no direct data connection between them.

Part of the complexity of workflow design is when the user needs to deal with collections, control structures such as iterations and error handling. Scuff is simplified to the extent that these are implicit. This layer fills in these implicit assumptions by interpreting an Internal Object Model that encodes the data that passes through a workflow. This data model is lightweight; it contains some basic data structures such as lists and trees, and enables the decoration of data with MIME types and semantic descriptions to enable later discovery or viewing of the data.

The addition of data structures such as lists to the data object model brings about an added complexity. There are a number of ways in which the list could be handled by the service. Taverna uses an implicit, but configurable, iteration mechanism as shown in Figure 20.5. Where a processor takes a single list as inputs, the enactment engine will invoke the processor multiple times and collate the results into a new list. Where a processor takes two (or more) list inputs, the service will be invoked with either the cross or dot product of the two lists.

Taverna supports fault tolerance through a configurable mechanism; processors will retry a failed service invocation a number of times, often with increasing delays between retry attempts before, finally, reporting failure. Users can specify alternative services for any Scuff processor in the order they should be substituted. Alternative services are typically either an identical service

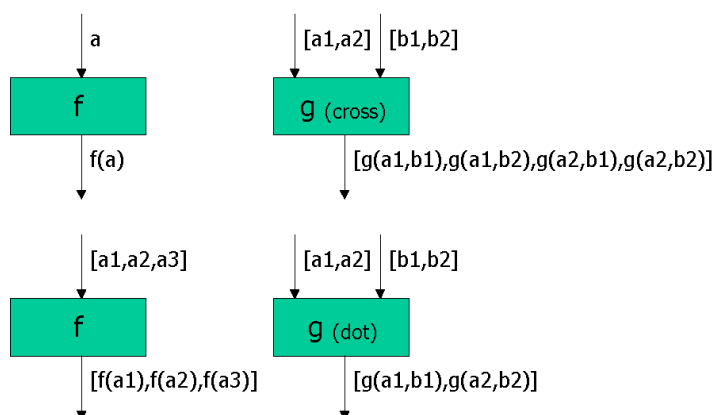


Fig. 20.5: Configurable iteration. For example, processor implements a function  $f$  - it takes one input  $a$  and produces result  $f(a)$ . If this processor is given a list of inputs  $[a1,a2,a3]$ , the implicit iteration will produce a list of results, one for each input. This is equivalent to 'map  $f$   $[a1,a2,a3]$ '. Where a processor has more than one input the default is to apply the function to the cross product of all the input lists, however sometimes the dot product is what is required. The configurable iterators allow users to specify how the lists of input value should be combined using these cross and dot operators.

supplied by an alternative service provider or, rarely, a completely different service that the user deems to be substitutable without damaging the workflow's intention.

Whilst the Scuf language defines the data flow, it does not fully describe the service interactions to enable this data flow.

It would be impossible to describe the interaction with all of the different service interfaces within a language like Scuf. Instead, Scuf is designed to be extensible through the use of processor types. We define a set of *Processor plug-ins* that manage service interaction by presenting a common abstraction over these different styles. Current processors include:

- A WSDL Scuf processor implemented by a single Web Service operation described in a WSDL file.
- A local Java function processor, where services are provided directly through a Java implementation with parameters as input ports and results as output ports (Fig. 20.1).
- A Soaplab processor, implemented through a CORBA-like stateful protocol of the Web Service operations in a Soaplab service.
- A nested workflow processor, implemented by a Scuf workflow (Figure 20.1).
- A BioMOBY processor (Fig. 20.6). Several smaller groups have adopted the BioMOBY project's conventions for publishing Web Services. Bio-

MOBY provides a registry and messaging format for bioinformatics services [440].

- A SeqHound Processor that manages a Representational State Transfer (REST) style interface, where all information required for the service invocation is encoded in a single HTTP GET or POST request (Fig. 20.6).
- A BioMart processor that directly accesses predefined queries over a relational database using a JDBC connection (Fig. 20.6).
- A Styx processor that executes a workflow subgraph containing streamed services using peer to peer data transfer based on the Styx Grid service protocol [338].

The Freefluo engine is responsible for the enactment of the workflow. The core of the engine is workflow language independent, with specific extensions that specialise Freefluo to enable it to enact Scuff.

## 20.5 Discovering resources and designing workflows

Workflow construction is driven by the domain expert, that is, the scientist. This corresponds to designing a suitable laboratory protocol for their investigation. The lifecycle of an *in silico* experiment (see Figure 20.2) has the following stages:

**Hypothesis formation:** First, the scientist determines the overall intention of the experiment. This informs a top-level design, and would be the overall “shape” of the workflow, including its inputs and desired outputs.

**Workflow design:** Second, this design is translated into a concrete plan. In the laboratory, this translation would consist of choosing appropriate experimental protocols and conditions. In an e-Science workflow, this maps to the choice and configuration of data and analysis services.

**Collecting:** The workflow needs to be run; the services invoked; data co-ordinated *etc.* (See Section 20.6). In the laboratory this is handled by protocols for entering results in laboratory books. As the workflow is executed, the results have to be collected and co-ordinated to record their derivation path. To comply with scientific practice, records need to be kept on where these data came from; when it was acquired; who designed and who ran the workflow and so forth. This is the provenance of the workflow and is described more fully in Section 20.7.

**Analysing and Sharing:** As in a laboratory experiment, results are analysed and then shared.

### 20.5.1 Service Discovery

In this section, we describe the *service discovery* and *service choice* aspects of running *in silico* experiments in Taverna.

Taverna uses a variety of different mechanisms for discovery of services, and populates the service list using an incremental approach. Flexible approaches to discovering available resources are an essential part of supporting the experimental lifecycle.

Public registries such as UDDI [402]. We are in favour of registries, but their limited usefulness is due to the lack of widespread deployment. They are generally perceived by the community to be a heavy-weight solution.

GRIMOIRES, an enriched prototype UDDI registry service developed by <sup>my</sup>Grid, with the ability to store semantic metadata about services.

URL submission: Users can add new services by directly pointing to a URL containing WSDL files. The workbench will introspect over the description and add the described services to a palette of services.

Workflow introspection: Users can exploit existing experience by loading existing workflows, observing how services have been used in context, and adding those services to the available services palette.

Processor-specific mechanisms: Many of the service types Taverna support through its processor plug-ins provide their own methods for Service discovery.

Scavenging: Local disks are scavenged for WSDL files that are introspected over, or users create a web page containing links to service descriptions and, when pointed at this page, Taverna explores all available service descriptions, extracts services and makes them available. While crude, this works well, and gives users considerable flexibility in loading the palette of available services that fit their current requirements.

Taverna's access to 3 000 services means that service selection is increasingly important. Figure 20.6) is grouped according to the service locations, which means that services of the same type are grouped together and colour-coded as in the workflow diagram (see figure 20.1), and supported by a simple search by name facility.

A common task is to locate a new service based on some conceptual description of the service semantics. To enable service selection by bioinformaticians, we must represent their view of the services and of the domain [449]. We have investigated a number of different mechanisms to drive the search process, including an RDF-based metadata enriched UDDI registry [270], and a domain ontology [450] described in the W3C Web Ontology Language OWL.

Feta is our third and most recent version of a component for semantically searching for candidate services that takes a user-oriented approach to service discovery [269], a path also being trodden by the BioMOBY project. In practice, this means we describe an abstraction over the services—provided by the Taverna processors—rather than the services themselves. We have relatively shallow descriptions of the services. Although richer descriptions might enable more refined searching and sophisticated reasoning they are expensive and time consuming to provide. In practice search results do not have to be precise, as the final choice is made by the workflow designer (a biologist), not

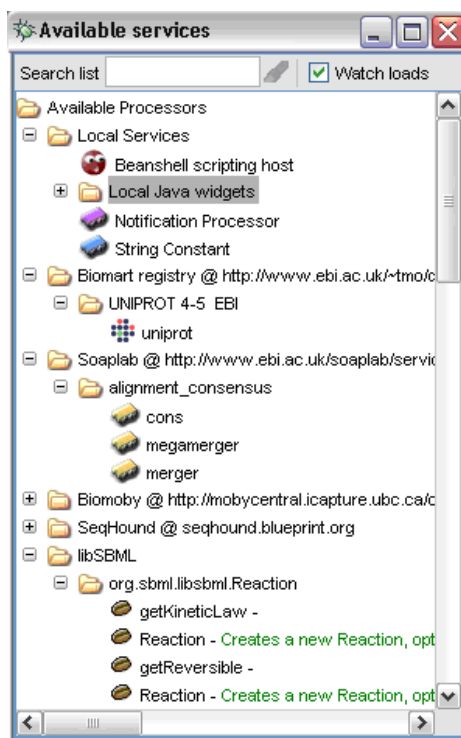


Fig. 20.6: An example palette of local (beanshell scripts, Java widgets) and remote (Biomart, Soaplab, Biomoby, Seqhound) services which can be used for the construction of workflows in Taverna. libSBML methods made available as local services via the API consumer and which were used for the construction of the exemplar systems biology workflow are also shown.

automatically by a machine. Finally, the use of shallow descriptions enables us to use simpler technologies to answer queries.

### 20.5.2 Service composition

Most workflow design packages have adopted a view analogous to electric circuit layout, with services represented as ‘chips’ with pins for input and output [68, 383]. However from a user interface point of view, this arrangement can become less understandable as complexity increases. If the layout of service components on screen is left under the user’s control then the user can tailor the workflow appearance, but this can result in a large amount of time being spent effectively doing graph layout rather than e-Science. In Taverna, the graphical view of a workflow is read-only; it is generated from the underlying workflow model. One advantage of this is that it is easy to

generate different graphical views of the workflow showing more or less detail as required.

When composing workflows in an open world we have no control over the data types used by the component services. A service, identified by a scientist as being suitable, may not use the same type as the preceding service in the workflow, even if the data matches at a conceptual level. Consequently, many of the bioinformatics workflows created in Taverna contain numerous ‘Shim’ services [217] that reconcile the inevitable type mismatches between autonomous third-party services. We are currently building libraries of shims for de-referencing identifiers, syntax and semantic translation, mapping, parsing, differencing and so on.

## 20.6 Executing and monitoring workflows

Execution of a workflow is largely an unseen activity, except for monitoring process and reviewing records of an experimental run (see Section 20.7). A critical requirement of <sup>my</sup>Grid’s service approach is that workflow invocation behaviour should be independent of the workflow enactment service used. To facilitate peer review of novel results, it is important that other scientists are able to reproduce *in silico* experiments in their context and verify that their results confirm the reported novel results.

Executing workflows using different enactment services is given less emphasis in business workflows, which will typically be carefully negotiated and agreed by the businesses involved, and executed in a fixed, known context. In contrast, a scientific workflow will be shared and evolved by a community and executed by many individual scientists using their favoured workflow enactment service.

### 20.6.1 Reporting

Reporting the progress of a workflow is a complex task. Information about service invocation is unavailable in the general case. Defining how far a service is through a given invocation, so progress can be displayed, is non-trivial without the explicit modelling and monitoring of state. The migration of application services to the Grid’s Web Service Resource Framework [239] is a solution that we are investigating.

The reporting mechanism in Taverna is a stream of events for each processing entity, with these events corresponding to state transitions of the service component. For example, a message is emitted when the service is first scheduled, when it has failed for the third time and is waiting to retry, *etc.* These message streams are collated into an XML document format and the results presented to the user in tabular form as shown in figure 20.7.

The introduction of reporting in Taverna does not alter the workflow results. What it does alter is users’ understanding of what is going on, and

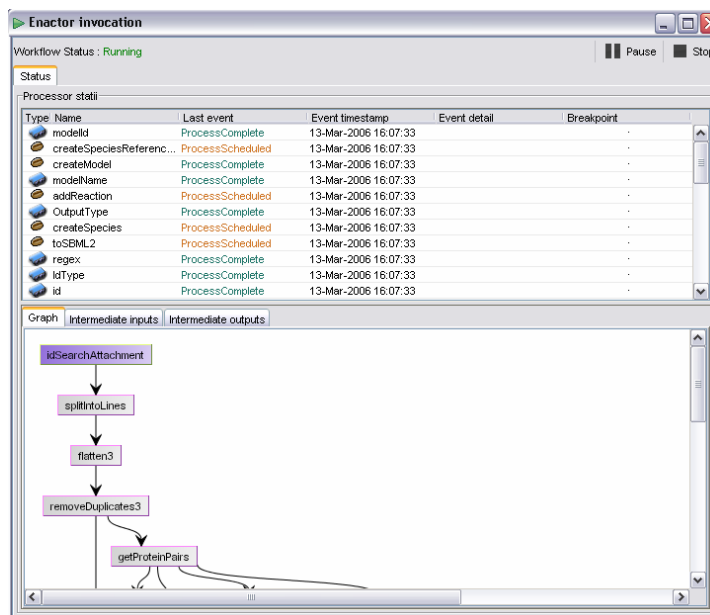


Fig. 20.7: Status information. When running a workflow the Taverna workbench displays status information from the workflow enactor. For each Scuff processor the last event is displayed along with the appropriate time and additional detail if available. This additional detail can include progress through an iteration (e.g. ‘item 2 of 6’) and retry information. The status information also allows the selection of a processor and viewing the relevant intermediate inputs and outputs. Each data item has been assigned an LSID. More detailed trace information is also available using the ‘Process report’ tab.

therefore their confidence that the system is doing what they want. Overall the feedback from Taverna’s initial users was that workflow execution without suitable monitoring was not acceptable. They were willing to accept workflows that occasionally failed; their experience with form-based web services was that these were unreliable. However, workflow execution could not be a ‘black-box’ service, users need feedback on what is happening, whether the workflow completed successfully or failed, and they need this recorded in logging records.

When a workflow may contain fifty or more processing components (e.g., Scuff processors), and each of these components can be retrying, using alternative implementations *etc.*, the complete state of a workflow is highly complex. Users require a visualization that allows them to see at a glance what is happening, acquire intermediate results where appropriate and control the workflow progress manually should that be required.

## 20.7 Managing and sharing workflows and their results

As the use of workflows increases, the ability to gather and generate data in large quantities, the storage of this data in an organised manner becomes essential for analysis within and between experiments. For scientists, workflows are the means to an end; their primary interest is in the results of experiments. This interest, however, goes beyond examining the results themselves and extends to the context within which those results exist. Specifically, the scientist will wish to know from where a particular result was derived, which key process was used and what parameters were applied to that process. Thus, in addition to the raw data, we have devised a model of metadata describing the provenance of all aspects of the experiment: the data's derivation path, an audit trail of the services invoked, the context of the workflow and the evidence of the knowledge outcomes as a result of its execution [457]. Another view is that it is the traditional who, where, when, what and how questions applied to *in silico* science. These different aspects of provenance can be used for life scientists in different scenarios:

- to repeat a workflow execution by retrieving the “recipe” recorded in the provenance;
- to reproduce a data product by retrieving the intermediate results or inputs that this data was derived from;
- to assess the performance of a service that is invoked in different experiment runs, at different times;
- to debug the failure of a workflow run, *e.g.*, which service failed, when and why it failed, *etc.*;
- to analyze the impacts of a service/database update on the experiment results, by comparing provenance of repeated runs;
- to “smartly” re-run a workflow if a service is updated, by using provenance to compute which part of a workflow is required to be re-run as a consequence of the update;
- to aggregate provenance of a common data product that is produced in multiple runs.

We have adopted two key technologies for provenance collection:

**Life Science Identifiers.** The description of the derivation of data necessitates reference to the data sets both inside and outside the control of <sup>m</sup>yGrid. Bioinformatics has adopted view standards for the identification of data, instead of using an *ad hoc* system of *accession numbers*. The recent Life Science Identifier (LSID) standard [127] provides a migration path from the legacy accession numbers, to an identification scheme based on URIs. **Resource Description Framework (RDF).** The Dako data store has a fixed schema that reflects the common entities used in e-Science experimental lifecycle untied to any scientific discipline. The use of a fixed schema provides performance benefits. However, RDF's basic graph data model



is well suited to the task of representing data derivation. The Knowledge Annotation and Verification of Experiments (KAVE) metadata store has a flexible schema due to its use of RDF. This allows statements to be added outside the fixed schema of the Dako data store, as is needed when providing subject specific information. KAVE enables other components in <sup>my</sup>Grid to store statements about resources and later query those statements.

One can distinguish between provenance of the data and provenance of the process, although the two are linked. The primary task for data provenance is to allow the exploration of results and the determination of the derivation path for the result itself in terms of input data and intermediate results *en route* to the final value. ‘Side effect’ information, about how intermediate and final results have been obtained, is generated and stored during workflow invocation. Thus the workflow engine produces not just results but also provenance metadata about those results. Side effect information is anything that could be recorded by some agent observing the workflow invocation, and it implicitly or explicitly links the inputs and outputs of each service operation within the workflow in some meaningful fashion. The associated component RDF-Provenance-Plug-in listens to the events of workflow execution and stores relevant statements using KAVE, for example, a name for a newly created data item or a meaningful link between the output of a service and the inputs that were used in its creation.

Process provenance is somewhat simpler than data provenance, and is similar to traditional event logging. Knowledge provenance is the most advanced and contextual of the metadata results. Often a user does not need to see a full ‘blow by blow’ account of the processes that executed during the workflow, or a full account of the complete data derivation path. Instead they wish to relate data outcomes across a group of processes annotating the relationships between outcomes with more semantically meaningful terms than ‘derived by’. As each such provenance fingerprint is unique to the workflow and the user, a *provenance template* accompanies the Scuff document to be populated by the provenance capture component and stored in the KAVE.

## 20.8 Related work

In Life Sciences there are many scientists who want an easy way of rapidly pulling together third-party services into prototypical *in silico* experiments. This contrasts with fields, such as physics and astronomy, where the prime scenario involves carefully designed workflows linking applications to exploit computational grid resources for *in silico* experiments that were previously impractical due to resource constraints.

Scientific workflow systems vary in terms of their intended scientific scope (the kinds of analyses supported), their technical scope (the kinds of resources

that can be composed), their openness to incorporating new services, and whether or not they are open source. The strengths of Taverna are its ability to link together a significant range of autonomous bioinformatics services and its flexibility, particularly in terms of the metadata generated to help manage and share workflow results.

The Kepler workflow system [67, 68] has been developed for ecologists, geologists and biologists in and is built on Ptolemy II, a mature application from electrical engineering [40]. Kepler's strengths include its library of actors, which are mainly local applications, and its suite of directors that provide flexible control strategies for the composition of actors. The Triana [383] system was originally developed as a data analysis environment for a gravitational wave detection project. Like Taverna and Kepler, Triana is also dataflow oriented. It is aimed at CPU intensive applications, allowing scientists to compose their local applications and distribute the computation.

DiscoveryNet uses a proprietary workflow engine and all services are wrapped to conform to a standard tabular data model. DiscoveryNet scientific workflows are used to allow scientists to plan, manage, share and execute knowledge discovery and data analysis procedures [352]. In the Pegasus system [184] users provide a workflow template and artificial intelligence planning techniques are used to co-ordinate the execution of applications on a heterogeneous and changing set of computational resources. The emphasis is on the scheduling large numbers of jobs on a computational Grid, where there may be alternative strategies for calculating a user's result set.

The use of workflows for 'programming in the large' to compose web services has led to significant interest in a standard workflow language, with BPEL<sup>1</sup> [70] a strong candidate, created through the agreed merge of IBM's WSFL [260] and Microsoft's XLANG [389]. One reason why Taverna workflows use Scuff rather than a potential standard is historical. In the initial stages of the <sup>my</sup>Grid project in 2001, BPEL did not exist. The more significant reason is conceptual. Initial experiments showed IBM's WSFL language did not match how our target users wanted to describe their *in silico* experiments [55]. WSFL forced users to think in terms of web service ports and messages rather than passing data between bio-services.

## 20.9 Discussion and Future Directions

<sup>my</sup>Grid set out to build a workflow environment to allow scientists to perform their current bioinformatics tasks in a more explicit, repeatable, and shareable manner.

---

<sup>1</sup> BPEL was originally termed BPEL4WS and is being promoted as a standard called WSBPEL through OASIS (Organization for the Advancement of Structured Information Standards), an international consortium for e-business standards.

Making tacit procedural knowledge explicit: For at least the last 250 years this has been recognised as essential in science. Each experiment must carry with it a detailed “methods” description, to allow others both to validate the results, but also re-use the experimental method. Our experience suggests that workflows allow this to be achieved for *in silico* experiments. They are formal, precise and explicit, yet straightforward to explain to others.

Ease of automation: Many of the analyses we support are already undertaken by scientists who orchestrate their applications by hand. Workflows can drastically reduce analysis time by automation. For example, Taverna workflows developed by the Williams Beuren Syndrome team have reduced a manual task that took 2 weeks to be an automated task that typically takes just over two hours [371].

Appropriate level of abstraction: Bioinformaticians have traditionally automated analyses through the use of scripting languages such as PERL. These are notoriously difficult to understand often because they can conflate the high level orchestration at the application level with low level “plumbing”.

Taverna and the myGrid suite enables users to rapidly *interoperate* services. It does not support the semantic *integration* of the data outcomes of those services. We underestimated the amount of data integration and visualisation provided by the existing web delivered applications. They often integrate information from many different analysis tools, and provide cross-references to other resources. Accessing the analysis tool directly as a service, circumvents this useful functionality. Although the scientist is presented with results in hours not weeks, it now takes a significant time to analyse the large amount of often fragmented results. A solution is complicated by the fact that the workflow environment does not “understand” the data, and so cannot perform the data integration necessary. We have provided integration steps within workflows, written as scripts that integrate and render results, but these are specific to each workflow design. We are currently investigating a multi-pronged approach: (i) the use of semantic web technology to provide more generic solutions that can be re-used between related workflows; (ii) appropriate workflow designs using Shims and services under the control of the user to build data objects; and (iii) closing off the open world in situations where the workflows are known to orchestrate a limited number of services and will be permanent in nature, so it is worth the effort of building a more strongly typed model.

From January 2006 the myGrid suite, including Taverna 2.0, moved to a new phase. As part of the UK’s Open Middleware Infrastructure Institute (OMII-UK)( <http://www.omii.ac.uk>) myGrid is to be integrated with a range of Grid services and deployed in a common container with job submission services, monitoring services and large-scale data management services. Focus is placed on the following:

- Grid deployment. Deploying the Taverna architecture within a Grid container, making the enactor a stateful service, and a server-side distributed service, and supporting stateful data repositories.
- Improved security. Authentication and authorisation management for data, metadata and implementation of credentials for access control of services.
- Revised execution and processor models to support interactive applications, long running processes, control-based workflows, data flows with large data throughput, enhanced provenance collection, and credential handling. We already have a user interaction service that allows users to participate interactively with workflows.
- Improved data and metadata management incorporating better user-oriented result viewers and incorporating SRB and OGSA-DAI data implementations.
- Integration with third party platforms such as with Toolbus and EGEE. We also plan to continue to interoperate with other workflow systems, specifically Kepler and the ActiveBPEL system emerging from UCL.
- Extending services that can be executed to more domain services such as the R suite, and to generic services such as GridSAM job submission.

The field of scientific workflows is rapidly evolving, and as a project in this area <sup>my</sup>Grid must also evolve. We engage different user communities (such as biological simulation), new applications become available, as do novel service frameworks for deploying them. By working closely with our users, service providers, and other workflow projects, we continue to extend the basic core functionality to fulfil a wide range of uses.

### Acknowledgements

This work is supported by the UK e-Science programme myGrid grants EP-SRC GR/R67743, EP/D044324/1, EP/C536444/1 and the Link-Up e-Science sisters project. The authors would like to acknowledge the <sup>my</sup>Grid team. Peter Li is funded by the BBSRC. Hannah Tipney developed workflows for investigation of Williams-Beuren Syndrome and is supported by The Wellcome Foundation (G/R:1061183). We also thank our industrial partners: IBM, Sun Microsystems, GlaxoSmithKline, AstraZeneca, Merck KgaA, geneticXchange, Epistemics Ltd, and Cerebra.



---

## Index

- autonomous service providers, 303
- bioinformatician, 299, 303
- bioinformatics, 299–318
- biological data, 299
- BioMOBY project, 301
- BPEL, 316
- collections, 306
- Condor, 303
- control flow, 305
- control structures, 306
- data flow, 303, 308
- derivation path, 309
- DiscoveryNet, 300, 303, 316
- distributed databases, 301
- Enactor Internal Object Model, 305
- e-Science, 301–318
- event-observer interface, 306
- execution layer, 304
- experimental protocol, 300
- experimental run, 312
- fault tolerance, 303, 308
- Feta, 310
- Freefluo, 300, 304, 305, 309
- GRIMOIRES, 310
- GriPhyN, 300
- heterogeneous interfaces, 304
- high-level presentation, 304
- implicit iteration, 304
- in silico* experiment, 299–318
  - validation, 312
- intermediate results, 314
- iteration, 306, 307
- KAVE, 315
- Kepler, 316
- layered architecture, 304
- Life Science Identifiers, 314
- middleware, 300
- MIME types, 307
- multi-tiered approach, 304
- <sup>my</sup>Grid, 299–318, *see also* Taverna
- <sup>my</sup>Grid architecture, 303–309
- OASIS, 316
- open world, 303, 312
  - service assumption, 303
- PathPort, 301
- Pegasus, 316
- Pegasus workflows, 303
- port types, 304
- Processor-specific mechanisms, 310
- provenance, 303, 314–315
  - collection, 314
  - data, 315
  - process, 315
  - use, 314
- Ptolemy II, 316
- Public registries, 310
- resource constraints, 316
- Resource Description Framework (RDF), 314
- result context, 314
- result derivation, 314
- result validation, 314
- SBML, 301
- SCEC/IT, 300
- scientific practice, 309
- Scufl, 304, 306, 308, 316
- SEEK, 300, 303
- Semantic descriptions, 307
- service

- conceptual description, 310
- service autonomy, 299
- service choice, 310
- service composition, 312
- service discovery, 309, 310
- service heterogeneity, 299
- service interactions, 308
- service invocation, 312
- service registry, 310
- service semantics, 310
- service substitution, 308
- Service-Orientated Architecture, 303
- shallow semantic description, 311
- Shim services, 312
- systems biology, 301
- Taverna, 299–318
  - alternative services, 308
  - iteration mechanism, 307
  - processor plug-in architecture, 305
  - processor types, 306, 308
  - requirements, 303, 304
  - user interface, 312
  - workflow, 301
  - Workflow Object Model, 306
- Triana, 316
- type system, 299
- URL submission, 310
- workflow
  - data-centric, 301
  - requirements, 303
  - Taverna, 301
  - workflow design, 309–312
  - workflow introspection, 310
  - workflow layout, 312
  - workflow monitoring, 312
  - workflow progress, 312
  - WSBPEL, 316, *see also* BPEL
  - WSDL scavenging, 310
  - WSFL, 316
  - XLANG, 316

---

## References

1. AMD. <http://www.amd.com>.
2. Apples with apples: Numerical relativity comparisons and tests. See web site at: <http://www.ApplesWithApples.org>.
3. Astrophysics Simulation Collaboratory (ASC) home page. <http://www.ascportal.org>.
4. The Cactus Framework. See web site at: <http://www.cactuscode.org>.
5. Center for computation and technology at louisiana state university. See <http://www.cct.lsu.edu/>.
6. DFN-Verein project “*development of grid based simulation and visualization techniques*” (GRIKSL) home page. <http://www.griksl.org>.
7. Doe grids certificate authority. See <http://www.doe grids.org/>.
8. Einstein@Home Project. See web site at: <http://www.physics2005.org/events/einsteinathome/>.
9. Fixed mesh refinement with Carpet. <http://www.carpetcode.org/>.
10. Flexible image transport system.
11. The galactic legacy infrared mid-plane survey extraordinaire (glimpse). <http://www.astro.wisc.edu/sirtf/>.
12. Glimpse validation images. <http://www.astro.wisc.edu/sirtf/2massimages/2massimages.html>.
13. Grid adaptive computational engine (grace). <http://www.caip.rutgers.edu/~parashar/TASSL/Projects/GrACE/>.
14. Grid3+. See <http://www.ivdgl.org/grid2003>.
15. Gridcat: Osg ce catalog. <http://osg-cat.grid.iu.edu>.
16. GridLab: A Grid application toolkit and testbed project home page. <http://www.gridlab.org>.
17. Gsi-enabled openssh. See <http://grid.ncsa.uiuc.edu/ssh/>.
18. Gums. See <http://grid.racf.bnl.gov/GUMS/>.
19. Ibm db2. See <http://www-306.ibm.com/software/data/db2>.
20. Ibm websphere. See web site at <http://www-306.ibm.com/software/websphere/>.
21. Illinois BioGrid. <http://www.illinoisbiogrid.org/>.
22. Intel. <http://www.intel.com>.
23. international virtual data grid laboratory. See project web site at: <http://www.ivdgl.org>.
24. Iphas image gallery. <http://astro.ic.ac.uk/Research/Halpha/North/gallery.shtml>.
25. Iphas: the int h-alpha emission survey. <http://iapetus.phy.umist.ac.uk/IPHAS/iphase.html>.



26. Ligo data grid (ldg). <http://www.lsc-group.phys.uwm.edu/lscdatagrid>.
27. Ligo data replicator. LIGO Data Replicator. See <http://www.lsc-group.phys.uwm.edu/ldr>.
28. LIGO Home Page. <http://www.ligo.caltech.edu>.
29. Load sharing facility. See web site at: <http://accl.grc.nasa.gov/lsf/>.
30. Matt anderson. <http://relativity.phys.lsu.edu/postdocs/matt/>.
31. The montage project page. <http://montage.ipac.caltech.edu/docs/download.html>.
32. The montage project web page.
33. Montage version 1.7.x documentation and download. <http://montage.ipac.caltech.edu/docs/>.
34. Montage version 1.7.x. photometric and calibration accuracy. <http://montage.ipac.caltech.edu/docs/accuracy.html>.
35. Mopex, the spitzer science center mosaic engine. <http://ssc.spitzer.caltech.edu/postbcd/doc/mosaicer.pdf>.
36. The open science grid consortium. <http://www.opensciencegrid.org/>.
37. Owl web ontology language. <http://www.w3.org/TR/owl-features/>.
38. Penn State LIGO Data Processing Center. <http://ligo.aset.psu.edu>.
39. Portable batch system. See <http://www.openpbs.org/>.
40. Ptolemy II.  
See website at <http://ptolemy.eecs.berkeley.edu/ptolemyII>.
41. Reality grid project. <http://www.realitygrid.org/>.
42. SAMRAI: Structured Adaptive Mesh Refinement Application Infrastructure. <http://www.llnl.gov/CASC/SAMRAI/>.
43. Sun storedge sam-qfs. Sun StorEdge SAM-QFS. See <http://www.sun.com>.
44. The teragrid project. <http://www.teragrid.org/>.
45. The 2MASS Project. <http://www.ipac.caltech.edu/2mass>.
46. UWM LSC Group Medusa Cluster. <http://www.lsc-group.phys.uwm.edu/beowulf/medusa>.
47. WSRF:Lite - Perl Grid Services. <http://www.sve.man.ac.uk/Research/AtoZ/ILCT>.
48. XML Process Definition Language (XPDL). Technical report WFMCTC-1025, Workflow Management Coalition, Lighthouse Point, Florida, USA, 2002.
49. Instant-Grid – A Grid Demonstration Toolkit. <http://instant-grid.de/>, 2006.
50. B. Abbott et al. Analysis of ligo data for gravitational waves from binary neutron stars. *Phys. Rev.*, D69:122001, 2004.
51. B. Abbott et al. Search for gravitational waves from galactic and extra-galactic binary neutron stars. *Phys. Rev.*, D72:082001, 2005.
52. B. Abbott et al. Search for gravitational waves from primordial black hole binary coalescences in the galactic halo. *Phys. Rev.*, D72:082002, 2005.
53. B. Abbott et al. Search for gravitational waves from binary black hole inspirals in ligo data. *Phys. Rev.*, D73:062001, 2006.
54. A. Abramovici, W. E. Althouse, R. W. P. Drever, Y. Gursel, S. Kawamura, F. J. Raab, D. Shoemaker, L. Sievers, R. E. Spero, and K. S. Thorne. LIGO - The Laser Interferometer Gravitational-Wave Observatory. *Science*, 256:325–333, Apr. 1992.
55. M. Addis, J. Ferris, M. Greenwood, D. Marvin, P. Li, T. Oinn, and A. Wipat. Experiences with science workflow specification and enactment in bioinformatics. In *Proceedings of UK e-Science All Hands Meeting*, pages 459–467, 2003.

56. W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, and S. Tuecke. Data management and transfer in high-performance computational grid environments. *Parallel Computing*, 28(5):749–771, May 2002.
57. B. Allen. A  $\chi^2$  time-frequency discriminator for gravitational wave detection. *Phys. Rev.*, D71:062001, 2005.
58. B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton. Findchirp: An algorithm for detection of gravitational waves from inspiraling compact binaries. *Submitted to Phys. Rev. D.*, 2005.
59. G. Allen, D. Angulo, I. Foster, G. Lanfermann, C. Liu, T. Radke, E. Seidel, and J. Shalf. The Cactus Worm: Experiments with dynamic resource discovery and allocation in a grid environment. *Int. J. of High Performance Computing Applications*, 15(4), 2001. [http://www.cactuscode.org/Papers/IJSA\\_2001.pdf](http://www.cactuscode.org/Papers/IJSA_2001.pdf).
60. G. Allen, D. Angulo, T. Goodale, T. Kielmann, A. Merzky, J. Nabrzysky, J. Pukacki, M. Russell, T. Radke, E. Seidel, J. Shalf, and I. Taylor. Grid-Lab: Enabling Applications on the Grid. In M. Parashar, editor, *GRID 2002, 3rd International Workshop on Grid Computing*, volume 2536, pages 39–45. Springer-Verlag, Jan 2002.
61. G. Allen, W. Bengler, T. Dramlitsch, T. Goodale, H. Hege, G. Lanfermann, A. Merzky, T. Radke, and E. Seidel. Cactus grid computing: Review of current development. In R. Sakellariou, J. Keane, J. Gurd, and L. Freeman, editors, *Europar 2001: Parallel Processing, Proceedings of 7th International Conference Manchester, UK August 28-31, 2001*. Springer, 2001. <http://www.cactuscode.org/Papers/Europar01.ps.gz>.
62. G. Allen, T. Goodale, G. Lanfermann, T. Radke, D. Rideout, and J. Thornburg. *Cactus Users Guide*, 2004. <http://www.cactuscode.org/Guides/Stable/UsersGuide/UsersGuideStable.pdf>.
63. M. Alpdemir, A. Mukherjee, A. Gounaris, N. Paton, A. Fernandes, R. Sakellariou, P. Watson, and L. P. Using OGSA-DQP to Support Scientific Applications for the Grid. In *Proc. First International Workshop on Scientific Applications in Grid Computing (SAG'04)*, pages 13–24. Springer-Verlag, LNCS, Vol. 3458, 2004. Invited Paper.
64. I. Alphaworks. Virtual XML Garden. <http://www.alphaworks.ibm.com/tech/virtualxml>, December 2005.
65. M. Alt, H. Bischof, and S. Gorlatch. Algorithm design and performance prediction in a Java-based Grid system with skeletons. In B. Monien and R. Feldmann, editors, *Euro-Par 2002*, volume 2400 of *Lecture Notes in Computer Science*, pages 899–906. Springer-Verlag, Aug. 2002.
66. M. Alt, A. Hoheisel, H.-W. Pohl, and S. Gorlatch. A Grid Workflow Language Using High-Level Petri Nets. In R. W. et al., editor, *Proceedings of the 6-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2005*, Poznan, Poland, 2005.
67. I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludäscher, and S. Mock. Kepler: An Extensible System for Design and Execution of Scientific Workflows. In *16th Intl. Conference on Scientific and Statistical Database Management (SSDBM)*, pages 423–424. IEEE Computer Society, 2004.
68. I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludäscher, and S. Mock. Kepler: Towards a Grid-enabled system for scientific workflows. In *Proceedings of the Workflow in Grid Systems Workshop at the Global Grid Forum (GGF10)*, 2004.

69. R. P. Anderson, D. Lew, and A. T. Peterson. Evaluating predictive models of species' distributions: Criteria for selecting optimal models. *Ecological Modelling*, 162:211–232, 2003.
70. T. Andrews, F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic, and S. Weerawarana. Business Process Execution Language for Web Services Version 1.1.
71. M. B. Araujo, M. Cabeza, W. Thuiller, L. Hannah, and P. H. Williams. Would climate change drive species out of reserves? An assessment of existing reserve-selection methods. *Global Change Biology*, 10:1618–1626, 2004.
72. M. B. Araujo, R. G. Pearson, W. Thuiller, and M. Erhard. Validation of species-climate impact models under climate change. *Global Change Biology*, 2005. in press.
73. A. Arbre, P. Avery, D. Bourilkov, R. Cavanaugh, S. Katageri, J. Rodriguez, G. Graham, J. Vöckler, and M. Wilde. Virtual Data in CMS Productions. Technical report, GriPhyN, 2003.
74. O. G. S. Architecture. <https://forge.gridforum.org/projects/ogsa-wg>.
75. R. Armstrong, D. Gannon, A. Geist, K. Keahey, S. Kohn, L. McInnes, S. Parker, and B. Smolinski. Toward a Common Component Architecture for High-Performance Scientific Computing. In *Proceedings of High Performance Distributed Computing (HPDC99)*, 1999.
76. D. Arnold, S. Agrawal, S. Blackford, J. Dongarra, M. Miller, K. Seymour, K. Sagi, Z. Shi, and S. Vadhiyar. Users' Guide to NetSolve V1.4.1. Technical Report ICL-UT-02-05, University of Tennessee, Knoxville, jun 2002.
77. M. E. Assessment. Ecosystem and human well-being reports. Island Press, Washington, D.C., 2005. <http://www.millenniumassessment.org>.
78. A. Baker. *The Secret Ways of Al Baker*. The Miracle Factory, 2003.
79. K. Ballinger, D. Ehnebuske, M. Gudgin, M. Nottingham, and P. Yendluri. WS-I Basic Profile Version 1.0. <http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html>, April 2004.
80. M. Bardeen, E. Gilbert, T. Jordan, P. Nepywoda, E. Quigg, M. Wilde, and Y. Zhao. The QuarkNet/Grid Collaborative Learning e-Lab. In *Workshop on Collaborative and Learning Applications of Grid Technology and Grid Education, CCGRID2005*, 2005.
81. B. C. Barish and R. Weiss. Ligo and the detection of gravitational waves. *Phys. Today*, 52 (Oct)(10):44–50, 1999.
82. P. Barman, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In *SOSP 2003*, September 2003.
83. J. E. Barnes and P. Hut. A hierarchical  $O(N \log N)$  force-calculation algorithm. *Nature*, 324(4):446–449, 1986.
84. C. Baru, A. Gupta, B. Ludäscher, R. Marciano, Y. Papakonstantinou, P. Veliikhov, and V. Chu. XML-Based Information Mediation with MIX. In *ACM Intl. Conference on Management of Data (SIGMOD)*, pages 597–599, Philadelphia, PA, 1999.
85. C. Baru, R. Moore, A. Rajasekar, and M. Wan. The SDSC storage resource broker. In *Proceedings of CASCON*, 1998.
86. M. Beckerle and M. Westhead. GGF DFDL Primer. Technical report, Global Grid Forum, 2004.

87. D. J. Beerling, B. Huntley, and J. P. Bailey. Climate and the distribution of *Fallopia japonica*: Use of an introduced species to test the predictive capacity of response surfaces. *Journal of Vegetation Science*, 6:269–282, 1995.
88. R. A. Benjamin et al. First glimpse results on the stellar structure of the galaxy. *Astrophysical Journal*, 600:L149–L152, 2005.
89. J. Bent, V. Venkataramani, N. LeRoy, A. Roy, J. Stanley, A. Arpaci-Dusseau, R. Arpaci-Dusseau, and M. Livny. NeST - a grid enabled storage appliance. In J. Weglarz, J. Nabrzyski, J. Schopf, and M. Stroinski, editors, *Grid Resource Management*. Kluwer Academic Publishers, 2003.
90. D. Berkley, S. Bowers, M. Jones, B. Ludäscher, M. Schildhauer, and J. Tao. Incorporating semantics in scientific workflow authoring. In *Proceedings of the International Conference on Scientific and Statistical Database Management (SSDBM)*, 2005.
91. L. Bernardinello and F. de Cindio. A survey of basic net models and modular net classes. In *Advances in Petri Nets 1992, The DEMON Project*, volume 609 of *LNCS*, pages 304–351, London, UK, 1992. Springer-Verlag.
92. M. Bevers, J. Hof, D. W. Uresk, and G. L. Schenbeck. Spatial optimization of prairie dog colonies for black-footed ferret recovery. *Operations Research*, 45:495–507, 1997.
93. Basic Local Alignment Search Tool (BLAST). <http://www.ncbi.nlm.nih.gov/blast/>, 2006.
94. J. Blythe, S. Jain, E. Deelman, Y. Gil, K. Vahi, A. Mandal, and K. Kennedy. Task Scheduling Strategies for Workflow-based Applications in Grids. In *CC-Grid*, Cardiff, UK, 2005.
95. Biological Magnetic Resonance Data Bank. <http://www.bmrb.wisc.edu/>, 2006.
96. L. Bocchi, C. Laneve, and G. Zavattaro. A calculus for long running transactions. In E. Najm, U. Nestmann, and P. Stevens, editors, *FMOODS 2003*, number 2884 in *Lecture Notes in Computer Science*, pages 124–138, 2003.
97. R. Bondarescu, G. Allen, G. Daues, I. Kelley, M. Russell, E. Seidel, J. Shalf, and M. Tobias. The Astrophysics Simulation Collaboratory portal: a framework for effective distributed research. *Future Generation Computer Systems*, 21(2):259–270, 2005.
98. D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, and D. Orchard. Web Services Architecture, W3C Working Group Note. <http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/>, November 2004.
99. S. Bowers and B. Ludäscher. An Ontology-Driven Framework for Data Transformation in Scientific Workflows. In *International Workshop on Data Integration in the Life Sciences (DILS'04)*, pages 1–16. Springer-Verlag, LNCS, Vol. 2994, 2004.
100. S. Bowers, D. Thau, R. Williams, and B. Ludäscher. Data procurement for enabling scientific workflows: On exploring inter-ant parasitism. *Lecture Notes in Computer Science*, 3372:57–63, 2005.
101. F. Breg, S. Diwan, J. Villacis., J. Balasubramanian, E. Akman, and D. Gannon. Java rmi performance and object model interoperability: Experiments with java/hpc++ distributed components. In *Concurrency Practice and Experience, Special Issue from the Fourth Java for Scientific Computing Workshop*, 1998.
102. D. A. Brown. Using the inspiral program to search for gravitational waves from low-mass binary inspiral. *Class. Quant. Grav.*, 22:S1097–S1108, 2005.

103. M. Brune, G. Fagg, and M. Resch. Message passing environments for meta-computing. *Future Generation Computer Systems*, 15(5-6):699–712, 1999.
104. D. Bunting et al. Web Services Context (WS-Context) Ver1.0. [http://www.arjuna.com/library/specs/ws\\_caf\\_1-0/WS-CTX.pdf](http://www.arjuna.com/library/specs/ws_caf_1-0/WS-CTX.pdf), 2003.
105. A. Buonanno, Y. Chen, and M. Vallisneri. Detecting gravitational waves from precessing binaries of spinning compact objects: Adiabatic limit. *Phys. Rev. D*, 67:104025, 2003.
106. A. Buonanno, Y. Chen, and M. Vallisneri. Detection template families for gravitational waves from the final stages of binary-black-hole inspirals: Nonspinning case. *Phys. Rev. D*, 67:024016, 2003.
107. C. E. Burns, K. M. Johnston, and O. J. Schmitz. Global climate change and mammalian species diversity in US national parks. *Proceedings of the National Academy of Sciences of the United States of America*, 100(20):11474–11477, 2003.
108. M. Butler, R. Pennington, and J. A. Terstriep. Mass storage at NCSA: SGI DMF and HP UniTree. In *Proceedings of 40th Cray User Group Conference*, 1998.
109. R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch. *IEEE Computer*, 33(12):60–66, 2000.
110. K. Camarda, Y. He, and K. A. Bishop. A parallel chemical reactor simulation using cactus. In *Proceedings of Linux Clusters: The HPC Revolution, NCSA*, 2001.
111. F. Camilo, D. Lorimer, P. Freire, A. Lyne, and R. Manchester. Observations of 20 millisecond pulsars in 47 tucanae at 20 cm. *The Astrophysical Journal*, (535):975, 2000.
112. J. Cao, S. A. Jarvis, S. Saini, and G. R. Nudd. Gridflow: Workflow management for grid computing. In *3rd International Symposium on Cluster Computing and the Grid*. IEEE Computer Society Press, 2003.
113. E. Caron, B. Del-Fabbro, F. Desprez, E. Jeannot, and J.-M. Nicod. Managing Data Persistence in Network Enabled Servers. *Scientific Programming Journal*, 2005.
114. E. Caron, F. Desprez, F. Lombard, J.-M. Nicod, M. Quinson, and F. Suter. A Scalable Approach to Network Enabled Servers. In *8th International EuroPar Conference ()*, volume 2400 of *LNCS*, pages 907–910, Paderborn, Germany, Aug. 2002. Springer-Verlag.
115. G. Carpenter, A. N. Gillison, and J. Winter. DOMAIN: A flexible modeling procedure for mapping potential distributions of animals and plants. *Biodiversity and Conservation*, 2:667–680, 1993.
116. H. Casanova, A. Legrand, D. Zagorodnov, and F. Berman. Heuristics for Scheduling Parameter Sweep Applications in Grid Environments. In *9th Heterogeneous Computing Workshop (HCW)*, pages 349–363, Cancun, may 2000.
117. CERN Advanced Storage Manager. <http://castor.web.cern.ch/castor/>, 2005.
118. Convention on Biodiversity Article 2 (Rio Earth Summit), 1992. <http://www.biodiv.org/convention/default.shtml>.
119. CCA Forum. The Common Component Architecture Technical Specification - version 0.5. Technical report, Common Component Architecture Forum, 2001.
120. CDDL Working Group, GGF. <https://forge.gridforum.org/projects/cddl-wg>.

121. L. Chen, S. J. Cox, F. Tao, N. R. Shadbolt, C. Puleston, and C. Goble. Empower resource providers to build the semantic grid. In *IEEE/WIC/ACM International Conference on Web Intelligence*, September 2004.
122. A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunszt, M. Ripeanu, B. Schwartzkopf, H. Stockinger, K. Stockinger, and B. Tierney. Giggle: A Framework for Constructing Scalable Replica Location Services. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–17. IEEE Computer Society Press, November 2002.
123. J. Chin, P. V. Coveney, and J. Harting. The teragyroid project: Collaborative steering and visualisation in an hpc grid for modelling complex fluids. *UK All-hands e-Science Conference, 2004*, Sept. 2004.
124. J. Chin, J. Harting, S. Jha, P. Coveney, A. R. Porter, and S. M. Pickles. Steering in computational science: mesoscale modelling and simulation. *Contemporary Physics*, 44:417–434, 2003.
125. D. Churches, G. Gombas, A. Harrison, J. Maassen, C. Robinson, M. Shields, I. Taylor, and I. Wang. Programming Scientific and Distributed Workflow with Triana Services. *Grid Workflow 2004 Special Issue of Concurrency and Computation: Practice and Experience*, To be published:1532–0634, 2005.
126. E. Churchwell et al. Rcw 49 at mid-infrared wavelengths: A glimpse from the spitzer space telescope. *The Astrophysical Journal Supplement Series*, 154:322–327, 2004.
127. T. Clark, S. Martin, and T. Liefeld. Globally distributed object identification for biological knowledgebases. *Briefings in Bioinformatics*, 5(1):59–70, 2004.
128. T. O. S. Coalition. OWL-S semantic markup for web services. <http://www.daml.org/services/owl-s/1.1/>, 2004.
129. J. Cohen, N. Furmento, G. Kong, A. Mayer, S. Newhouse, and J. Darlington. RealityGrid: An Integrated Approach to Middleware through ICENI. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences (to appear)*, aug 2005.
130. J. Cohen, W. Lee, A. Mayer, and S. Newhouse. Making the Grid Pay - Economic Web Services. In *Building Service Based Grids Workshop, GGF11*, Honolulu, Hawaii, USA, June 2004.
131. U. H. P. Computing and Communication. <http://www.usc.edu/hpcc>.
132. Condor glidein. <http://www.cs.wisc.edu/condor/glidein>.
133. Condor Team. DAGMan: A Directed Acyclic Graph Manager, July 2005. <http://www.cs.wisc.edu/condor/dagman/>.
134. M. Corp. Web Services Specifications Index. <http://msdn.microsoft.com/webservices/understanding/specs/>.
135. J. Costa, A. T. Peterson, and C. B. Beard. Ecological niche modeling and differentiation of populations of *Triatoma brasiliensis* Neiva, 1911, the most important Chagas disease vector in northeastern Brazil (Hemiptera, Reduviidae, Triatominae). *American Journal of Tropical Medicine & Hygiene*, 67:516–520, 2002.
136. K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *10th IEEE International Symposium on High Performance Distributed Computing*, pages 181–184. IEEE Press, 2001.

137. K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke. A Resource Management Architecture for Metacomputing Systems. In *Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing*, pages 62–82. IEEE Computer Society, 1998.
138. B. D. and C. F. (ed). Web Services Addressing 1.0 - Core (WS-Addressing). Technical report, W3C, 2005.
139. DAGMan (Directed Acyclic Graph Manager). Web Page.
140. H. Daily, H. Casanovay, and F. Berman. A Decoupled Scheduling Approach for the GrADS Program Development Environment. In *Proceedings of the Supercomputing 2002 conference*, Baltimore, November 2002.
141. J. Darlington. Parallel programming using skeleton functions. *Notes in Computer Science*, 694:146–160, 1993.
142. S. Das, A. McGough, J. Cohen, and J. Darlington. Lightweight Solution for Protein Annotation. In *UK e-Science All Hands Meeting*, pages 396–402, Nottingham, UK, Sept. 2005. ISBN 1-904425-53-4.
143. dCache.org. <http://www.dcache.org/>, 2006.
144. E. Deelman, J. Blythe, Y. Gil, and C. Kesselman. Workflow Management in GriPhyN. In J. Nabrzyski, J. Schopf, and J. Weglarz, editors, *Grid Resource Management*. Kluwer, 2003.
145. E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, S. Patil, M.-H. Su, K. Vahi, and M. Livny. Pegasus : Mapping Scientific Workflows onto the Grid. In *2nd European Across Grids Conference*, Nicosia, Cyprus, 2004.
146. E. Deelman, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, K. Blackburn, A. Lazzarini, A. Arbree, R. Cavanaugh, and S. Koranda. Mapping Abstract Complex Workflows onto Grid Environments. *Journal of Grid Computing*, 1(1):25–39, 2003.
147. E. Deelman, C. Kesselman, G. Mehta, L. Meshkat, L. Pearlman, K. Blackburn, P. Ehrens, A. Lazzarini, R. Williams, and S. Koranda. Griphyn and ligo, building a virtual data grid for gravitational wave scientists. In *HPDC*, pages 225–, 2002.
148. E. Deelman, G. Mehta, and C. Kesselman. Transformation Catalog Design for GriPhyN. Technical Report GriPhyN-2001-17, Information Sciences Institute, 2001.
149. E. Deelman, R. Plante, C. Kesselman, G. Singh, M.-H. Su, G. Greene, R. Hanisch, N. Gaffney, A. Volpicelli, J. Annis, V. Sekhri, T. Budavari, M. A. Nieto-Santisteban, W. O'Mullane, D. Bohlender, T. McGlynn, A. H. Rots, and O. Pevunova. Grid-based galaxy morphology analysis for the national virtual observatory. In *SC*, page 47, 2003.
150. E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, and D. Katz. Pegasus: a framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming Journal*, 13(2), November 2005.
151. T. Delaitre, T. Kiss, A. Goyeneche, G. Terstyanszk, S. Winter, and P. Kacsuk. GEMICA: Running Legacy Code Applications as Grid Services. *Journal of Grid Computing*, 3(1-2):75–90, 2005.
152. Apache Derby. Web Page.
153. F. Dijkstra and A. van der Steen. Integration of Two Ocean Models. In *Special Issue of Concurrency and Computation, Practice & Experience*. Wiley, 2005. (In Press).

154. M. J. Duftler, N. K. Mukhi, A. Slominski, and S. Weerawarana. Web Services Invocation Framework (WSIF). In *OOPSLA 2001 Workshop on Object-Oriented Web Services*, 2001.
155. L. Dutka, B. Kryza, K. Krawczyk, M. Majewska, R. Slota, L. Hluchy, and J. Kitowski. Component-expert Architecture for Supporting Grid Workflow Construction Based on Knowledge. In P. C. P and M. Cunningham, editors, *Innovation and the Knowledge Economy. Issues, Applications, Case Studies*, volume 2, pages 239–246. IOS Press, 2005.
156. J. D. Eckart and B. Sobral. A Life Scientists Gateway to Distributed Data Management and Computing: The PathPort/ToolBus Framework. *OMICS: A Journal of Integrative Biology*, 7(1):79–88, 2003.
157. Eclipse. Graphical editing framework.  
See web site at: <http://www.eclipse.org/gef/>.
158. Eclipse. Standard widget toolkit.  
See web site at: <http://www.eclipse.org/swt/>.
159. EGEE: Enabling Grids for E-science in Europe.  
See website at <http://public.eu-egee.org/>.
160. J. Eker, J. Janneck, E. Lee, J. Liu, X. Liu, J. Ludvig, S. Neuendorffer, S. Sachs, and Y. Xiong. Taming Heterogeneity – the Ptolemy Approach. In *Proceedings of the IEEE*, volume 91(1), pages 127–144, January 2003.
161. J. Elith and M. Burgman. Predictions and their validation: Rare plants in the Central Highlands, Victoria. In J. Scott, P. J. Heglund, and M. L. Morrison, editors, *Predicting Species Occurrences: Issues of Scale and Accuracy*. Island Press, Washington, D.C., 2002.
162. W. Emmerich, B. Butchart, L. Chen, B. Wassermann, and S. L. Price. Grid service orchestration using the business process execution language (bpel). *Journal of Grid Computing*, 2005.
163. D. Epema, M. Livny, R. van Dantzig, X. Evers, and J. Pruyne. A worldwide flock of Condors: Load sharing among workstation clusters. *Future Generation Computer Systems*, 12:53–65, 1996.
164. M. H. Eres, G. E. Pound, Z. Jiao, J. L. Wason, F. Xu, A. J. Keane, and S. J. Cox. Implementation of a grid-enabled problem solving environment in matlab. In *International Conference on Computational Science*, pages 420–429, 2003.
165. Etnus. Totalview.
166. T. Fahringer, J. Qin, and S. Hainzer. Specification of grid workflow applications with agwl: An abstract grid workflow language. In *International Symposium on Cluster Computing and the Grid*. IEEE Computer Society Press, May 2005.
167. A. Faulkener, I. Stairs, M. Kramer, A. Lyne, G. Hobbs, A. Possenti, D. Lorimer, R. Manchester, M. McLaughlin, N. D’Amico, F. Camilo, and M. Burgay. The parkes multibeam pulsar survey: V. finding binary and millisecond pulsars. *Mon. Not. R. Astron. Soc.*, (355):147–159, 2004.
168. E. Field, T. Jordan, and C. Cornell. Opensha: A community-modeling environment for seismic hazard research. *Seismological Research Letters*, 74(4):406–419, 2003.
169. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – HTTP/1.1. *Internet RFC 2616*, Jun 1999.
170. G. G. Forum. Global Grid Forum.  
See web site at: <http://www.gridforum.org/>.



171. I. Foster and C. Kesselman. Globus: A Metacomputing Infrastructure Toolkit. *Int. Journal of Supercomputing Applications*, 11(2):115–128, 1997.
172. I. Foster and C. Kesselman, editors. *The Grid: Blueprint for a New Computer Infrastructure*. Morgan-Kaufmann, 1999.
173. I. Foster, C. Kesselman, J. Nick, and S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Technical report, Open Grid Service Infrastructure WG, Global Grid Forum, 2002.
174. I. Foster, C. Kesselman, and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organization. *The International Journal of High Performance Computing Applications*, 15(3):200–222, Fall 2001.
175. I. Foster, J. Voeckler, M. Wilde, and Y. Zhao. Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation. In *14th International Conference on Scientific and Statistical Database Management (SSDBM'02)*, pages 37–46, Edinburgh, Scotland, 2002.
176. I. Foster, J. Voeckler, M. Wilde, and Y. Zhao. The Virtual Data Grid: A New Model and Architecture for Data-intensive Collaboration. In *Conference on Innovative Data Systems Research*, January 2003.
177. Fraunhofer-Gesellschaft. The Fraunhofer Resource Grid. <http://www.fhrg.fraunhofer.de/>, 2006.
178. J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke. Condor-G: A computation management agent for multi-institutional grids. *Cluster Computing*, 5:237–246, 2002.
179. J. Frey, T. Tannenbaum, M. Livny, I. Foster, and S. Tuecke. Condor-G: A Computation Management Agent for Multi-Institutional Grids. In *Proceedings of the 10th IEEE International Symposium on High Performance Distributed Computing (HPCD-'01)*, 2001.
180. N. Furmento, W. Lee, A. Mayer, S. Newhouse, and J. Darlington. ICENI: an open grid service architecture implemented with Jini. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, pages 1–10, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.
181. M. Galperin. The Molecular Biology Database Collection: 2006 update. *Nucleic Acids Research*, 34(Database issue):3–5, Jan 2006.
182. E. Gamma and K. Beck. *Contributing to eclipse: Principles, Patterns, and Plug-Ins*. Addison-Wesley, 2004.
183. GEO 600 aims at the direct detection of gravitational waves . <http://www.geo600.uni-hannover.de/>.
184. Y. Gil, E. Deelman, J. Blythe, C. Kessleman, and H. Tangmunarunkit. Artificial Intelligence and Grids: Workflow Planning and Beyond. *IEEE Intelligent Systems special issue on e-science*, 19(1):26–33, 2004.
185. T. Glatard, J. Montagnat, and X. Pennec. An optimized workflow enactor for data-intensive grid applications. Technical Report 05.32, I3S Laboratory, Sophia Antipolis, France, <http://www.i3s.unice.fr/~glatard/publis/RR-05.32-T.GLATARD.pdf>, Oct. 2005.
186. T. Glatard, J. Montagnat, and X. Pennec. Grid-enabled workflows for data intensive applications. In *Computer Based Medical Systems (CBMS'05), special track on Grids for Biomedicine and Bioinformatics*, Dublin, Ireland, June 2005.
187. gLite middleware. <http://www.gLite.org>.
188. The Globus Alliance. See website at <http://www.globus.org>.

189. T. Goodale, G. Allen, G. Lanfermann, J. Masso, T. Radke, E. Seidel, and J. Shalf. The cactus framework and toolkit: Design and applications. In *Vector and Parallel Processing VECPAR 2002, 5th International Conference, Lecture Notes in Computer Science*. Springer, 2003.
190. T. Goodale, I. Taylor, and I. Wang. Integrating Cactus Simulations within Triana Workflows. In *Proceedings of 13th Annual Mardi Gras Conference - Frontiers of Grid Applications and Technologies*, pages 47–53. Louisiana State University, February 2005.
191. J. Grethe, C. Baru, A. Gupta, M. James, B. Ludäscher, M. Martone, P. Papadopoulos, P. ST., A. Rajasekar, S. Santini, I. Zaslavsky, and M. Ellisman. Biomedical Informatics Research Network: Building a national collaboratory to hasten the derivation of new understanding and treatment of disease. *Stud Health Technol Inform*, 112:100–109, 2005.
192. Grid ENabled Integrated Earth system model project. <http://www.genie.ac.uk/>.
193. J. Grinnell. Field tests of theories concerning distributional control. *American Naturalist*, 51:115–128, 1917.
194. J. Grinnell. Geography and evolution. *Ecology*, 5:225–229, 1924.
195. W. Gropp, E. Lusk, N. Doss, and A. Skjellum. A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing*, 22(6):789–828, Sept. 1996.
196. J. S. D. L. W. Group. <https://forge.gridforum.org/projects/jsdl-wg>.
197. J. Grundy, M. Apperley, J. Hosking, and W. Mugrdige. A decentralized architecture for software process modeling and enactment. *IEEE Internet Computing*, 2(5):53 – 62, 1998.
198. J. Grundy, J. Hosking, R. Amor, W. Mugrdige, and M. Li. Domain-specific visual languages for specifying and generating data mapping system. *Journal of Visual Languages and Computing*, 15(3-4):243–263, June - August 2004.
199. T. Gubala, M. Bubak, M. Malawski, and K. Rycerz. Semantic-based Grid Workflow Composition. In R. W. et al., editor, *Proceedings of the 6-th Intl. Conf. on Parallel Processing and Applied Mathematics PPAM'2005*, Poznan, Poland, 2005.
200. M. Y. Gulamali, A. S. McGough, R. J. Marsh, N. R. Edwards, T. M. Lenton, P. J. Valdes, S. J. Cox, S. J. Newhouse, J., and Darlington. Performance guided scheduling in genie through iceni. In *Proceedings of the UK e-Science All Hands Meeting 2004*, Nottingham, September 2004.
201. W. W. Hargrove, R. H. Gardner, M. G. Turner, W. H. Romme, and D. G. Despain. Simulating fire patterns in heterogeneous landscapes. *Ecological Modelling*, 135:243–263, 2000.
202. A. Harrison and I. Taylor. Wspeer - an interface to web service hosting and invocation. In *HIPS Joint Workshop on High-Performance Grid Computing and High-Level Parallel Programming Models*, 2005.
203. J. Hau, W. Lee, and J. Darlington. A semantic similarity measure for semantic web services. *Web Service Semantics: A workshop at The Fourteen International World Wide Web Conference (WWW2005)*, 2005.
204. R. L. Henderson. Job scheduling under the portable batch system. In *JSSPP*, pages 279–294, 1995.
205. S. I. Higgins, D. M. Richardson, R. M. Cowling, and T. H. Trinder-Smith. Predicting the landscape-scale distribution of alien plants and their threat to plant diversity. *Conservation Biology*, 13:303–313, 1999.

206. S. Hinz, K. Schmidt, and C. Stahl. Transforming BPEL to Petri Nets. In W. van der Aalst, B. Benatallah, F. Casati, and F. Curbera, editors, *Proceedings of the Third International Conference on Business Process Management (BPM 2005)*, volume 3649 of *Lecture Notes in Computer Science*, pages 220–235, Nancy, France, September 2005. Springer-Verlag.
207. C. Hoare. *Communicating Sequential Processes*. Prentice Hall International, 1985.
208. A. Hoheisel and U. Der. An XML-Based Framework for Loosely Coupled Applications on Grid Environments. In P. M. A. Sloot, D. Abramson, A. V. Bogdanov, J. J. Dongarra, A. Y. Zomaya, and Y. E. Gorbachev, editors, *Computational Science – ICCS 2003*, volume 2657 of *LNCS*, pages 245–254. Springer-Verlag, 2003.
209. A. Hoheisel and U. Der. Dynamic workflows for grid applications. In *Proceedings of the Cracow Grid Workshop '03*, Cracow, Poland, 2003.
210. J. R. Holden and H. Ammon. Prediction of possible crystal structures for c-, h-, n-, o-, and f-containing organic compounds. *Journal of Computational Chemistry*, 14(4):422–437, 1993.
211. R. D. Holt. Adaptive evolution in source-sink environments: Direct and indirect effects of density-dependence on niche evolution. *Oikos*, 75:182–192, 1996.
212. R. D. Holt and M. S. Gaines. Analysis of adaptation in heterogeneous landscapes: Implications for the evolution of fundamental niches. *Evolutionary Ecology*, 6:433–447, 1992.
213. G. Holzmann. *The SPIN model checker: Primer and reference manual*. Addison Wesley, 2003.
214. M. hoMe-made OpTimisEd scUfl enactoR. <http://www.i3s.unice.fr/~glatard>.
215. M. Hovestadt, O. Kao, A. Keller, and A. Streit. Scheduling in HPC Resource Management Systems: Queuing vs. Planning. *Lecture Notes in Computer Science*, 2862, Oct. 2003.
216. M. Hucka, A. Finney, H. Sauro, H. Bolouri, J. Doyle, H. Kitano, A. Arkin, B. Bornstein, D. Bray, A. Cornish-Bowden, A. Cuellar, S. Dronov, E. Gilles, M. Ginkel, V. Gor, I. Goryanin, W. Hedley, T. Hodgman, J. Hofmeyr, P. Hunter, N. Juty, J. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L. Loew, D. Lucio, P. Mendes, E. Minch, E. Mjolsness, Y. Nakayama, M. Nelson, P. Nielsen, T. Sakurada, J. Schaff, B. Shapiro, T. Shimizu, H. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, Mar 2003.
217. D. Hull, R. Stevens, P. Lord, C. Wroe, and C. Goble. Treating shimantic web syndrome with ontologies. In *First Advanced Knowledge Technologies workshop on Semantic Web Services (AKT-SWS04) KMi, The Open University, Milton Keynes, UK. 2004-12-08*, 2004. Workshop proceedings CEUR-WS.org ISSN:1613-0073.
218. L. M. Hunter and L. Rinner. The association between environmental perspective and knowledge and concern with species diversity. *Society & Natural Resources*, 17(6):517–532, 2004.
219. B. Huntley, P. J. Bartlein, and I. C. Prentice. Climatic control of the distribution and abundance of Beech (*Fagus L.*) in Europe and North America. *Journal of Biogeography*, 16:551–560, 1989.

220. B. Huntley, P. M. Berry, W. Cramer, and A. P. McDonald. Modelling present and potential future ranges of some European higher plants using climate response surfaces. *Journal of Biogeography*, 22:967–1001, 1995.
221. IBM. WSDL4J.  
See web site at: <http://sourceforge.net/projects/wsdl4j>.
222. IBM and BEA. BPELJ: BPEL for Java . <http://www.ibm.com/developerworks/webservices/library/ws-bpelj/>, 2004.
223. Immunology Grid, Immunology Grid Project. <http://www.immunologygrid.org>.
224. R. Irani and S. J. Bashna. *AXIS: Next Generation Java SOAP*. Wrox Press, may 2002.
225. ISO/IEC 10026-1. Information technology – Open Systems Interconnection – Distributed Transaction Processing – Part 1: OSI TP Model, 1998.
226. ISO/IEC 15909-1. High-level Petri nets – Part 1: Concepts, definitions and graphical notation, 2004.
227. ISO/IEC 15909-2. High-level Petri nets – Part 2: Transfer Format, 2005. Working Draft.
228. K. Jensen. An introduction to the theoretical aspects of Coloured Petri Nets. In J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *A Decade of Concurrency, Lecture Notes in Computer Science*, volume 803, pages 230–272. Springer-Verlag, 1994.
229. A. C. Jones, R. J. White, W. A. Gray, F. A. Bisby, N. Caithness, N. Pittas, X. Xu, T. Sutton, N. J. Fiddian, A. Culham, M. Scoble, P. Williams, O. Bromley, P. Brewer, C. Yesson, and S. Bhagwat. Building a biodiversity grid. In A. Konagaya and K. Satou, editors, *Grid Computing in Life Science: First International Workshop on Life Science Grid, Revised selected and invited papers (LNCS/LNBI 3370)*, pages 140–151. Springer Verlag, 2005.
230. A. C. Jones, X. Xu, N. Pittas, W. A. Gray, N. J. Fiddian, R. J. White, J. S. Robinson, F. A. Bisby, and S. M. Brandt. SPICE: a flexible architecture for integrating autonomous databases to comprise a distributed catalogue of life. pages 981–992.
231. M. Jones. SEEK EcoGrid: Integrating data and computational resources for ecology. *DataBits: An Electronic Newsletter for Information Managers, Long Term Ecological Research Program*, 2003.
232. T. Jordan and P. Maechling. The seec community modeling environment - an information infrastructure for system-level earthquake science. *Seismological Research Letters*, 74(3):324–328, 2003.
233. M. Jünger, E. Kindler, and M. Weber. The Petri Net Markup Language. In S. Philippi, editor, *7. Workshop Algorithmen und Werkzeuge für Petrinetze*, pages 47–52, <http://www.informatik.hu-berlin.de/top/pnml/>, 2000. Universität Koblenz-Landau.
234. Project JXTA, July 2005. <http://www.jxta.org>.
235. P. Kacsuk, A. Goyeneche, T. Delaitre, T. Kiss, Z. Farkas, and T. Boczko. High-Level Grid Application Environment to Use Legacy Codes as OGSA Grid Services. In *Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04)*, pages 428–435, 2004.
236. P. Kacsuk and G. Sipos. Multi-Grid, Multi-User Workflows in the P-GRADE Grid Portal. *Journal of Grid Computing*, 3(3-4), 2005.
237. V. Kalogera et al. The cosmic coalescence rates for double neutron star binaries. *Ap. J.*, 601:L179–L182, 2004.

238. G. Kandaswamy, L. Fang, Y. Huang, S. Shirasuna, S. Marru, and D. Gannon. Building web services for scientific grid applications. *IBM Journal of Research and Development*, 50, 2006.
239. Karl Czajkowski et al. The WS-Resource Framework. <http://www.globus.org/wsrif>.
240. D. S. Katz, J. C. Jacob, G. B. Berriman, J. Good, A. C. Laity, E. Deelman, C. Kesselman, G. Singh, M.-H. Su, and T. A. Prince. A comparison of two methods for building astronomical image mosaics on a grid. In *34th International Conference on Parallel Processing Workshops ICPP 2005 Workshops, 14-17 June 2005, Oslo, Norway*, pages 85–94. IEEE Computer Society, 2005.
241. K. Keahey, K. Doering, and I. Foster. From Sandbox to Playground: Dynamic Virtual Environments in the Grid. In *5th IEEE/ACM International Workshop on Grid Computing*, Nov. 2004.
242. J. Kim, Y. Gil, and M. Spraragen. A Knowledge-Based Approach to Interactive Workflow Composition. In *14th International Conference on Automatic Planning and Scheduling (ICAPS 04)*, Whistler, Canada, 2004.
243. J. Kim, M. Spraragen, and Y. Gil. An Intelligent Assistant for Interactive Workflow Composition. In *Proceedings of the 2004 International Conference on Intelligent User Interfaces (IUI)*, January 2004.
244. D. King and A. Dennis. Gridant: A client-controllable grid workflow system. In *Proc. for the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 7*. IEEE Computer Society Press, 2004.
245. K. Knight and D. Marcu. Machine Translation in the Year 2004. In *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.
246. S.-H. Ko, K. W. Cho, Y. D. Song, and Y. G. Kim. Development of Cactus Driver for CFD Analyses in the Grid Computing Environment. In *Advances in Grid Computing: EGC 2005 (Lecture Notes in Computer Science)*, volume 3470, 2005.
247. G. Kola, T. Kosar, and M. Livny. A client-centric grid knowledgebase. In *Proceedings of 2004 IEEE International Conference on Cluster Computing*, pages 431–438, San Diego, CA, September 2004. IEEE.
248. G. Kola, T. Kosar, and M. Livny. Run-time adaptation of grid data-placement jobs. *Parallel and Distributed Computing Practices*, 2004.
249. G. Kola and M. Livny. Diskrouter: A flexible infrastructure for high performance large scale data transfers. Technical Report CS-TR-2003-1484, University of Wisconsin-Madison Computer Sciences Department, 2003.
250. T. Kosar. *Data Placement in Widely Distributed Systems*. PhD thesis, University of Wisconsin-Madison, 2005.
251. T. Kosar and M. Livny. Stork: Making data placement a first class citizen in the grid. In *Parallel and Distributed Computing Practices*, 2004.
252. S. Krishnan, P. Wagstrom, and G. von Laszewski. GSFL: A Workflow Framework for Grid Services. In *Preprint ANL/MCS-P980-0802*, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, U.S.A., 2002.
253. D. Kuo, P. Greenfield, S. Parastatidis, and J. Webber. Rules-based SSDL Protocol Framework. <http://www.ssd1.org/docs/v1.3/html/Rules%20SSDL%20Protocol%20Framework%20v1.3.html>, April 2005.
254. LCG2 middleware. <http://lcg.web.cern.ch/LCG/activities/~middleware.html>.

255. Linked Environments for Atmospheric Discovery. <http://lead.ou.edu/>.
256. LEAD Year-2 Annual Report. [http://lead.ou.edu/pdfs/LEAD\\_Year-2\\_Report.pdf](http://lead.ou.edu/pdfs/LEAD_Year-2_Report.pdf), 2005.
257. W. Lee, A. McGough, and J. Darlington. Performance Evaluation of the GridSAM Job Submission and Monitoring System. In *UK e-Science All Hands Meeting*, pages 915–922, Nottingham, UK, Sept. 2005. ISBN 1-904425-53-4.
258. W. Lee, S. McGough, S. Newhouse, and J. Darlington. A standard based approach to job submission through web services. In S. Cox, editor, *Proc. of the UK e-Science All Hands Meeting, Nottingham*, pages 901–905. UK EPSRC, 2004. ISBN 1-904425-21-6.
259. R. S. Levine, M. Q. Benedict, and A. T. Peterson. Distribution of *Anopheles quadrimaculatus* Say s.l. and implications for its role in malaria transmission in the US. *Journal of Medical Entomology*, 41:607–613, 2004.
260. F. Leyman. Web Services Flow Language (WSFL) 1.1. Technical report, IBM Software Group, 2001.
261. A. Lin, L. Dai, J. Mock, S. Peltier, and M. Ellisman. The telescience tools: Version 2.0. *Proceedings of The 1st IEEE International Conference on e-Science and Grid Computing*, pages 56–63, 2005.
262. A. Lin, L. Dai, K. Ung, S. Peltier, and M. Ellisman. The telescience project: Applications to middleware interaction components. *Proceedings of The 18th IEEE International Symposium on Computer-Based Medical Systems*, pages 543–548, 2005.
263. M. Litzkow, M. Livny, and M. Mutka. Condor - a hunter of idle workstations. In *Proceedings of the 8th International Conference on Distributed Computing Systems*, pages 104–111, June 1988.
264. London e-Science Centre. A Market for Computational Services. Available at <http://www.lesc.ic.ac.uk/markets/>.
265. C. J. Lonsdale et al. Swire: The sirtf wide-area infrared extragalactic survey. *Pub Ast Soc Pac*, 115:897, 2003.
266. M. Lorch and D. Kafura. Symphony — A Java-based Composition and Manipulation Framework for Computational Grids. In *Proceedings of the CC-Grid2002*, Berlin, Germany, 2002.
267. H. D. Lord. Improving the Application Environment with Modular Visualization Environments. *Computer Graphics*, 29(2):10–12, 1995.
268. P. Lord, P. Alper, C. Wroe, and C. Goble. Feta: A light-wieght architecture for user oriented semantic service discovery. In A. Gómez-Pérez and J. Euzenat, editors, *European Semantic Web Conference*, pages 17–31. Springer-Verlag, 2005.
269. P. Lord, S. Bechhofer, M. D. Wilkinson, G. Schiltz, D. Gessler, D. Hull, C. Goble, and L. Stein. Applying Semantic Web Services to bioinformatics: Experiences gained, lessons learnt. In *3rd International Semantic Web Conference (ISWC2004)*, pages 350–364. Springer-Verlag, LNCS, Vol. 3298, 2004.
270. P. Lord, C. Wroe, R. Stevens, C. Goble, S. Miles, L. Moreau, K. Decker, T. Payne, and J. Papay. Semantic and Personalised Service Discovery. In W. K. Cheung and Y. Ye, editors, *WI/IAT 2003 Workshop on Knowledge Grid and Grid Intelligence*, pages 100–107, Oct 2003.
271. D. Lorimer and M. Kramer. *A Handbook of Pulsar Astronomy*. Cambridge University Press, Cambridge, 2005.
272. F. S. E. Ltd. Failure Divergence Refinement: FDR2 User Manual. <http://www.fs.el.com/documentation/fdr2/>.

273. B. Ludäscher, I. Altintas, C. Berkley, D. G. Higgins, E. Jaeger, M. Jones, E. A. Lee, and Y. Zhao. Scientific workflow management and the kepler system. concurrency and computation: Practice and experience. *Concurrency and Computation: Practice and Experience, Special Issue on Scientific Workflows*, 2006. to appear.
274. S. A. Ludwig, W. Naylor, J. Padget, and O. F. Rana. Matchmaking support for mathematical web services. In *Proceedings of the UK e-Science All Hands Meeting 2005*, Nottingham, September 2005.
275. A. G. Lyne and G. Smith. *Pulsar Astronomy, 3rd Edition*. Cambridge University Press, Cambridge, 2005.
276. P. Maechling, H. Chalupsky, M. Dougherty, E. Deelman, Y. Gil, S. Gullapalli, V. Gupta, C. Kesselman, J. Kim, G. Mehta, B. Mendenhall, T. Russ, G. Singh, M. Spraragen, G. Staples, and K. Vahi. Simplifying construction of complex workflows for non-expert users of the southern california earthquake center community modeling environment. *ACM SIGMOD Record*, 34(3):24–30, 2005.
277. P. Maechling, V. Gupta, N. Gupta, E. Field, D. Okaya, and T. Jordan. Grid computing in the sceec community modeling environment. *Seismological Research Letters*, 76(5):518–587, 2005.
278. P. Maechling, V. Gupta, N. Gupta, E. Field, D. Okaya, and T. Jordan. Hazard map calculations using grid computing. *Seismological Research Letters*, 76(5):565–573, 2005.
279. M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley Series in Parallel Computing. John Wiley and Sons, 1995. <http://www.di.unito.it/~greatspn/GSPN-Wiley/>.
280. E. Martínez-Meyer. *Evolutionary Trends in Ecological Niches of Species*. PhD thesis, University of Kansas, 2002.
281. E. Martínez-Meyer, A. T. Peterson, and W. W. Hargrove. Ecological niches as stable distributional constraints on mammal species, with implications for Pleistocene extinctions and climate change projections for biodiversity. *Global Ecology and Biogeography*, 13:305–314, 2004.
282. A. Mayer, S. McGough, N. Furmento, J. Cohen, M. Gulamali, L. Young, A. Afzal, S. Newhouse, and J. Darlington. ICENI: An Integrated Grid Middleware to Support e-Science. In V. Getov and T. Kielmann, editors, *Component Models and Systems for Grid Applications*, volume 1 of *CoreGRID series*, pages 109–124. Springer, June 2004.
283. A. Mayer, S. McGough, M. Gulamali, L. Young, J. Stanton, S. Newhouse, and J. Darlington. Meaning and behaviour in grid oriented components. *Lecture Notes in Computer Science*, 2536:100–111, jan 2002.
284. A. E. Mayer. *Composite Construction of High Performance Scientific Applications*. PhD thesis, Department of Computing, Imperial College, London, UK, 2001.
285. D. McDermott. Estimated-Regression Planning for Interactions with Web Services. In *AI Planning Systems Conference*, 2002.
286. A. McGough, L. Young, A. Afzal, S. Newhouse, and J. Darlington. Workflow Enactment in ICENI. In *UK e-Science All Hands Meeting*, pages 894–900, Nottingham, UK, Sept. 2004. ISBN 1-904425-21-6.
287. S. McGough, A. Afzal, J. Darlington, N. Furmento, A. Mayer, and L. Young. Making the Grid Predictable through Reservations and Performance Modelling. *The Computer Journal*, 48(3):358–368, 2005.

288. S. McIlraith and T. Son. Adapting Golog for programming in the semantic web. In *Fifth International Symposium on Logical Formalizations of Commonsense Reasoning*, 2001.
289. J. O. Meyneke. Effects of global climate change on geographic distributions of vertebrates in North Queensland. *Ecological Modelling*, 174(4):347–357, 2004.
290. S. Microsystems. Java Remote Method Invocation (RMI). <http://java.sun.com/products/jdk/rmi/>.
291. L. Miles, A. Grainger, and O. Phillips. The impact of global climate change on tropical forest biodiversity in Amazonia. *Global Ecology and Biogeography*, 13(6):553–565, 2004.
292. R. Milner. *Communicating and Mobile Systems: the Pi-Calculus*. Cambridge University Press, 1999.
293. Microsoft Corporation. Windows Workflow Foundation. <http://msdn.microsoft.com/windowsvista/building/workflow/>, 2005.
294. S. Miyazaki, H. Sugawara, K. Ikeo, T. Gojobori, and Y. Tateno. DDBJ in the stream of various biological data. *Nucl. Acids Res.*, 32(90001):D31–34, 2004.
295. D. J. Mladenoff, T. A. Sickley, R. G. Haight, and A. P. Wydeven. A regional landscape analysis and prediction of favorable gray wolf habitat in the northern Great Lakes region. *Conservation Biology*, 9:279–294, 1995.
296. L. Moreau, Y. Zhao, I. Foster, J. Voekler, and M. Wilde. XDTM: XML Dataset Typing and Mapping for Specifying Datasets. In *European Grid Conference*, 2005.
297. N. Mulyar and W. van der Aalst. Patterns in Colored Petri Nets. In *BETA Working Paper Series*, WP 139. Eindhoven University of Technology, Eindhoven, 2005.
298. P. Murray-Rust. Chemical markup language. *World Wide Web Journal*, 2(4):135–147, 1997.
299. myLEAD. <http://www.cs.indiana.edu/dde/projects/lead.html>.
300. H. Nakada, S. Matsuoka, K. Seymour, J. Dongarra, C. Lee, and H. Casanova. A GridRPC Model and API for End-User Applications. Technical report, Global Grid Forum (GGF), jul 2005.
301. G. F. national grid initiative. <http://www.grid5000.org>.
302. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. Edutella: A p2p networking infrastructure based on rdf. In *11th World Wide Web Conference*, page 604, May 2002 2002.
303. NESC. National e-science centre.  
See web site at: <http://www.nesc.ac.uk/>.
304. H. A. Nix. A biogeographic analysis of australian elapid snakes. In R. Longmore, editor, *Atlas of elapid snakes of Australia*, pages 4–15. Canberra, Australian Government Publishing Service, 1986.
305. OASIS. OASIS Web Services Business Process Execution Language (WS-BPEL) TC. <http://www.oasis-open.org/committees/wsbpel>.
306. OASIS. Web Services Base Notification 1.3 (WS-BaseNotification). [http://docs.oasis-open.org/wsn/wsn-ws\\_base\\_notification-1.3-spec-pr-02.pdf](http://docs.oasis-open.org/wsn/wsn-ws_base_notification-1.3-spec-pr-02.pdf).
307. OASIS. Web Services Composite Application Framework (WS-CAF). <http://www.oasis-open.org/committees/ws-caf/charter.php>.
308. OASIS. Web services Context (WS-CTX). [www.iona.com/devcenter/standards/WS-CAF/WSCTX.pdf](http://www.iona.com/devcenter/standards/WS-CAF/WSCTX.pdf).



309. OASIS. Web Services Security (WS-Security). <http://www.oasis-open.org/committees/wss>.
310. OASIS. WS-Resource Properties (WSRF-RP), June 2004. <http://docs.oasis-open.org/wsrp/2004/06/wsrp-WS-ResourceProperties-1.2-draft-04.pdf>.
311. OASIS Open. Web Services Resource 1.2 (WS-Resource). [http://docs.oasis-open.org/wsrp/wsrp-ws\\_resource-1.2-spec-cs-01.pdf](http://docs.oasis-open.org/wsrp/wsrp-ws_resource-1.2-spec-cs-01.pdf).
312. Object Management Group (OMG). The Common Object Request Broker Architecture (CORBA). <http://www.corba.org>.
313. A. O'Brien, S. Newhouse, and J. Darlington. Mapping of scientific workflow within the e-protein project to distributed resources. In *Proceedings of the UK e-Science All Hands Meeting 2004*, Nottingham, September 2004.
314. T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li. Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows. *Bioinformatics*, 20(17):3045–3054, Nov 2004.
315. K. B. Olsen, S. M. Day, J. B. Minster, Y. Cui, A. Chourasia, M. Faerman, R. Moore, P. Maechling, and T. H. Jordan. Strong shaking in los angeles expected from southern san andreas earthquake. *Geol. Res. Lett.*, in press, 2006.
316. Open Grid Services Architecture Data Access and Integration (OGSA-DAI). <http://www.ogsadai.org.uk/>.
317. B. J. Owen. Search templates for gravitational waves from inspiraling binaries: Choice of template spacing. *Phys. Rev. D*, 53:6749–6761, 1996.
318. B. J. Owen and B. S. Sathyaprakash. Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement. *Phys. Rev. D*, 60:022002, 1999.
319. S. Panagiotidi, E. Katsiri, and J. Darlington. On Advanced Scientific Understanding, Model Componentisation and Coupling in GENIE. In *All Hands Meeting*, Nottingham, UK, September 2005.
320. S. Parastatidis and J. Webber. CSP SSDL Protocol Framework. <http://www.ssd1.org/docs/v1.3/html/CSP%20SSDL%20Protocol%20Framework%20v1.3.html>, April 2005.
321. S. Parastatidis and J. Webber. MEP SSDL Protocol Framework. <http://www.ssd1.org/docs/v1.3/html/MEP%20SSDL%20Protocol%20Framework%20v1.3.html>, April 2005.
322. S. Parastatidis and J. Webber. The SOAP Service Description Language. <http://www.ssd1.org/docs/v1.3/html/SSDL%20v1.3.html>, April 2005.
323. S. Parastatidis, J. Webber, S. Woodman, D. Kuo, and P. Greenfield. An Introduction to the SOAP Service Description Language v1.3. <http://www.ssd1.org/docs/v1.3/html/SSDL%20whitepaper%20v1.3.html>, April 2005.
324. S. Parastatidis, J. Webber, S. Woodman, D. Kuo, and P. Greenfield. Using SSDL to Describe and Reason about Asynchronous and Multi-Message Interactions between Web Services. *IEEE Internet Computing*, Jan/Feb 2006.
325. R. G. Pearson, T. P. Dawson, P. M. Berry, and P. A. Harrison. SPECIES: A spatial evaluation of climate impact on the envelope of species. *Ecological Modelling*, 154:289–300, 2002.
326. S. Peltier, A. Lin, D. Lee, S. Mock, S. Lamont, T. Molina, M. Wong, M. Martone, and M. Ellisman. The telescience portal for advanced tomography ap-

- lications. *Journal of Parallel and Distributed Applications*, 63(5):539–550, 2003.
327. D. Pennington and W. Michener. The EcoGrid and the Kepler workflow system: A new platform for conducting ecological analyses. *ESA Bulletin (Emerging Technologies)*, 86:169–176, 2005.
328. A. T. Peterson. Predicting the geography of species’ invasions via ecological niche modeling. *Quarterly Review of Biology*, 78:419–433, 2003.
329. A. T. Peterson and D. A. Kluza. New distributional modeling approaches for Gap Analysis. *Animal Conservation*, 6:47–54, 2003.
330. A. T. Peterson, E. Martínez-Meyer, C. González-Salazar, and P. Hall. Modeled climate change effects on distributions of Canadian butterfly species. *Canadian Journal of Zoology*, 82:851–858, 2004.
331. A. T. Peterson, M. A. Ortega-Huerta, J. Bartley, V. Sanchez-Cordero, J. Soberon, R. H. Buddemeier, and D. R. B. Stockwell. Future projections for Mexican faunas under global climate change scenarios. *Nature*, 416:626–629, 2002.
332. A. T. Peterson, J. Soberón, and V. Sanchez-Cordero. Conservatism of ecological niches in evolutionary time. *Science*, 285:1265–1267, 1999.
333. A. T. Peterson and D. A. Vieglais. Predicting species invasions using ecological niche modeling. *BioScience*, 51:363–371, 2001.
334. C. A. Petri. *Kommunikation mit Automaten*. PhD thesis, Institut für Instrumentelle Mathematik, Bonn, 1962.
335. S. J. Phillips, M. Dudik, and R. E. Schapire. A maximum entropy approach to species distribution modeling. In *Proceedings of the International Conference on Machine Learning*, 2004.
336. S. Pickles, J. Brooke, F. Costen, E. Gabriel, M. Müller, M. Resch, and S. Ord. Metacomputing across intercontinental networks. *Future Generation Computer Systems*, 17(5-6):911–918, 2001.
337. S. M. Pickles, P. V. Coveney, and B. M. Boghosian. Transcontinental realitygrids for interactive collaborative exploration of parameter space (triceps). Winner of SC’03 HPC Challenge Competition (Most Innovative Data-Intensive Application), November 2003.
338. R. Pike and D. M. Ritchie. The Styx Architecture for Distributed Systems. *Bell Labs Technical Journal*, 4(2):146–152, Apr-Jun 1999.
339. C. Pinchak, P. Lu, and M. Goldenberg. Practical heterogeneous placeholder scheduling in overlay metacomputers: Early experiences. In *JSSPP*, pages 205–228, 2002.
340. V. D. Pope, M. L. Gallani, V. J. Rowntree, and R. A. Stratton. The impact of new physical parametrizations in the Hadley Centre climate model – HadAM3. Technical report, Bracknell, Berks, UK, Hadley Centre for Climate Prediction and Research, 2002.
341. P.-G. portal. <http://www.lpds.sztaki.hu/pgportal/>.
342. J. Postel and J. Reynolds. File transfer protocol (FTP). *Internet RFC 959*, Oct 1985.
343. T. Pratt and M. Zelkowitz. *Programming Languages - Design and Implementation*. Prentice Hall, 3rd edition, 1999.
344. S. Price. The computational prediction of pharmaceutical crystal structures and polymorphism. *Advanced Drug Delivery Reviews*, 56(3):301–319, 2004.
345. T. J. C. Process. Portlet specification. Web Page.

346. R. Raman, M. Livny, and M. Solomon. Matchmaking: Distributed resource management for high throughput computing. In *Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing (HPDC7)*, Chicago, IL, July 1998.
347. W. Recommendation. Architecture of the World Wide Web, Volume One. <http://www.w3.org/TR/webarch>.
348. W. Reisig. *Elements Of Distributed Algorithms: Modeling and Analysis with Petri Nets*. Springer-Verlag, sep 1998.
349. W. Reisig. An Informal Introduction to Petri Nets - Running Example: A Producer/Consumer System. In *Petri Nets 2000, 21st International Conference on Application and Theory of Petri Nets, 2000*.
350. M. Resch, R. D., and S. R. Metacomputing experience in a transatlantic wide area application testbed. *Future Generation Computer Systems*, 15(5-6):807–816, 1999.
351. M. P. Robertson, N. Caithness, and M. H. Villet. A PCA-based modelling technique for predicting environmental suitability for organisms from presence records. *Diversity & Distributions*, 7:15–27, 2001.
352. A. Rowe, D. Kalaitzopoulos, M. Osmond, M. Ghanem, and Y. Guo. The Discovery Net system for high throughput bioinformatics. *Bioinformatics*, 19(90001):225i–231, 2003.
353. SAGA Research Group (GGF).  
See web site at: <https://forge.gridforum.org/projects/saga-rg/>.
354. E. Saxon, B. Baker, W. Hargrove, F. Hoffman, and C. Zganjar. Mapping environments at risk under different global climate change scenarios. *Ecology Letters*, 8(1):53–60, 2005.
355. Southern california earthquake center (scec). <http://www.scec.org/>.
356. Scec community modeling environment (scec/cme) project. <http://www.scec.org/cme>.
357. D. J. Schlegel, D. Finkbeiner, and M. Davis. Maps of dust infrared emission for use in estimation of reddening and cosmic microwave background radiation foregrounds. *Astrophysical Journal*, 500:525, 1998.
358. E. Schnetter, S. H. Hawley, and I. Hawke. Evolutions in 3d numerical relativity using fixed mesh refinement. *Class. Quantum Grav.*, 2003.
359. M. Senger. Bibliographic query service. <http://industry.ebi.ac.uk/openBQS/>, 2002.
360. M. Senger, P. Rice, and T. Oinn. Soaplab – a unified Sesame door to analysis tools. In *Proceedings of UK e-Science All Hands Meeting*, pages 509–513, September 2003.
361. S. Shirasuna. X Baya Workflow Composer. <http://www.extreme.indiana.edu/xgws/xbaya>.
362. G. Singh, S. Bharathi, A. L. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil, and L. Pearlman. A metadata catalog service for data intensive applications. In *SC*, page 33, 2003.
363. M. Skrutskie et al. The two micron all-sky survey(2mass). *Astronomy Journal*, 131(1163), 2006.
364. E. Smith and P. Anderson. Dynamic Reconfiguration for Grid Fabrics. In *5th IEEE/ACM International Workshop on Grid Computing*, Nov. 2004.
365. M. Snir, S. W. Otto, S. Huss-Lederman, D. W. Walker, and J. Dongarra. *MPI: The Complete Reference*. The MIT Press, Cambridge, MA, 1996.

366. Simple Object Access Protocol (SOAP) 1.2. Technical report, W3C, 2003.
367. J. Soberón and A. T. Peterson. Interpretation of models of fundamental ecological niches and species' distributional areas. *Biodiversity Informatics*, 2:1–10, 2005.
368. B. R. Stein and J. Wiecek. Mammals of the world: MaNIS as an example of data integration in a distributed network environment. *Biodiversity Informatics*, 1:14–22, 2004.
369. L. Stein. Creating a bioinformatics nation. *Nature*, 417:119 – 120, 2002.
370. R. Stevens, A. Robinson, and C. Goble. myGrid: Personalised Bioinformatics on the Information Grid. In *11th International Conference on Intelligent Systems in Molecular Biology*, volume 19 of *Bioinformatics*, pages i302–i304, June 2003.
371. R. Stevens, H. Tipney, C. Wroe, T. Oinn, M. Senger, P. Lord, C. Goble, A. Brass, and M. Tassabehji. Exploring Williams Beuren Syndrome Using myGrid. In *Bioinformatics*, volume 20, pages i303–310, 2004. Intelligent Systems for Molecular Biology (ISMB) 2004.
372. R. D. Stevens, A. J. Robinson, and C. A. Goble. mygrid: personalised bioinformatics on the information grid. In *ISMB (Supplement of Bioinformatics)*, pages 302–304, 2003.
373. D. Stockwell and I. R. Noble. Induction of sets of rules from animal distribution data: A robust and informative method of data analysis. *Mathematics and Computers in Simulation*, 33:385–390, 1992.
374. D. Stockwell and D. Peters. The GARP modelling system: problems and solutions to automated spatial prediction. *International Journal of Geographical Information Science*, 13(2):143–158, 1999.
375. D. Stockwell and D. Peters. The GARP modelling system: Problems and solutions to automated spatial prediction. *International Journal of Geographical Information Science*, 13:143–158, 1999.
376. A. S. Szalay, editor. *An Architecture for Access to a Compute Intensive Image Mosaic Service in the NVO.*, volume 4686. Proceedings of SPIE, 2002.
377. B. Talbot, S. Zhou, and G. Higgins. Review of the Cactus framework: Software engineering support of the third round of scientific grand challenge investigations, task 4 report - earth system modeling framework survey. [http://sdcd.gsfc.nasa.gov/ESS/esmf\\_tasc/Files/Cactus.b.html](http://sdcd.gsfc.nasa.gov/ESS/esmf_tasc/Files/Cactus.b.html).
378. Y. Tanaka, H. Nakada, S. Sekiguchi, T. Suzumura, and S. Matsuoka. Ninf-G: A Reference Implementation of RPC-based Programming Middleware for Grid Computing. *Journal of Grid Computing (JGC)*, 1(1):41–51, 2003.
379. I. Taylor, M. Shields, and R. Philp. GridOneD: Peer to Peer visualization using Triana: A Galaxy formation Test Case. In *UK eScience All Hands Meeting*, 2002.
380. I. Taylor, M. Shields, I. Wang, and A. Harrison. Visual Grid Workflow in Triana. *Journal of Grid Computing*, 3(3-4):153–169, September 2005.
381. I. Taylor, M. Shields, I. Wang, and A. Harrison. Visual Grid Workflow in Triana. *Journal of Grid Computing, Special Edition on Workflow*, To be published, 2006.
382. I. Taylor, M. Shields, I. Wang, and O. Rana. Triana Applications within Grid Computing and Peer to Peer Environments. *Journal of Grid Computing*, 1(2):199–217, 2003.

383. I. Taylor, I. Wang, M. Shields, and S. Majithia. Distributed computing with Triana on the Grid. *Concurrency and Computation: Practice and Experience*, 17(9):1197–1214, 2005.
384. I. J. Taylor, O. F. Rana, R. Philp, I. Wang, and M. S. Shields. Supporting Peer-2-Peer Interactions in the Consumer Grid. In *Eighth International Workshop on High-Level Parallel Programming Models and Supportive Environments (HIPS'03)*, pages 3–14. IEEE Computer Society, April 2003.
385. I. J. Taylor, M. S. Shields, I. Wang, and R. Philp. Distributed P2P Computing within Triana: A Galaxy Visualization Test Case. In *17th International Parallel and Distributed Processing Symposium (IPDPS 2003)*, pages 16–27. IEEE Computer Society, 2003.
386. Teragrid project. <http://www.teragrid.org>.
387. D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.
388. S. Thakkar, J. L. Ambite, and C. A. Knoblock. Composing, Optimizing, and Executing Plans for Bioinformatics Web services. *VLDB Journal, Special Issue on Data Management, Analysis and Mining for Life Sciences*, 14(3):330–353, Sep 2005.
389. S. Thatte. XLANG Web Services for Business Process Design. [http://www.gotdotnet.com/team/xml\\_wsspecs/xlang-c/default.htm](http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm), 2001.
390. The General Coupling Framework (GCF) approach. [http://www.cs.man.ac.uk/cnc-bin/cnc\\_gcf.pl](http://www.cs.man.ac.uk/cnc-bin/cnc_gcf.pl).
391. The GridSAM project. <http://gridsam.sourceforge.net>.
392. The K-Wf Grid Project. Knowledge-based Workflow System for Grid Applications. <http://www.kwfgrid.net/>, 2006.
393. The K-Wf Grid Project. The Grid Workflow Description Language Toolbox. <http://www.gridworkflow.org/kwfgrid/gworkflowdl/docs/>, 2006.
394. The Kerrighed project. <http://www.kerrighed.org/>.
395. I. The MathWorks. Matlab<sup>®</sup>. <http://www.mathworks.com/products/matlab/>.
396. The open Mosix project. <http://openmosix.sourceforge.net/>.
397. The open SSI project. <http://openssi.org/index.shtml>.
398. C. D. Thomas, A. Cameron, R. E. Green, M. Bakkenes, L. J. Beaumont, Y. C. Collingham, B. F. N. Erasmus, M. Ferreira de Siqueira, A. Grainger, L. Hannah, L. Hughes, B. Huntley, A. S. Van Jaarsveld, G. E. Midgely, L. Miles, M. A. Ortega-Huerta, A. T. Peterson, O. L. Phillips, and S. E. Williams. Extinction risk from climate change. *Nature*, 427:145–148, 2004.
399. K. S. Thorne. Gravitational radiation. In S. W. Hawking and W. Israel, editors, *Three hundred years of gravitation*, chapter 9, pages 330–458. Cambridge University Press, Cambridge, 1987.
400. W. Tim Berners-Lee. Web Architecture from 50,000 feet. <http://www.w3.org/DesignIssues/Architecture.html>.
401. The Triana Project.
402. UDDI Technical White Paper. Technical report, OASIS UDDI, September 2000.
403. T. UNICORE Forum. UNICORE: UNiform Interface to COmputing REsources. See website at <http://www.unicore.org>.
404. User Mode Linux. <http://user-mode-linux.sourceforge.net/>.

405. A. Vakali, B. Catania, and M. A. XML Data Stores: Emerging Practices. *Internet Computing*, 9(2):62–69, March/April 2005.
406. W. van der Aalst. Pi calculus versus Petri nets: Let us eat humble pie rather than further inflate the pi hype. [www.bptrends.com](http://www.bptrends.com), 2005.
407. W. van der Aalst and A. ter Hofstede. Workflow Patterns: On the Expressive Power of (Petri-net-based) Workflow Languages. Technical report, Department of Technology Management, Eindhoven University of Technology P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands, 2002.
408. R. A. Van Engelen and K. A. Gallivan. The gSOAP Toolkit for Web Services and Peer-to-Peer Computing Networks. In *Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '02)*, page 128, Washington, DC, USA, 2002. IEEE Computer Society.
409. J. Van Horn, J. Dobson, J. Woodward, M. Wilde, Y. Zhao, J. Voeckler, and I. Foster. Grid-Based Computing and the Future of Neuroscience Computation. In C. Senior, T. Russell, and M. S. Gazzaniga, editors, *Methods in Mind, Cognitive Neuroscience*. MIT Press (In Press), July 2006.
410. The virtual data toolkit. <http://www.vdt.org>.
411. B. Victor, F. Moller, M. Dam, and L. Eriksson. The Mobility workbench. <http://www.it.uu.se/research/group/mobility/mwb>.
412. J. E. Villacis, M. Govindaraju, D. Stern, A. Whitaker, F. Breg, P. Deuskar, B. Temko, D. Gannon, and R. Bramley. CAT: A high performance distributed component architecture toolkit for the grid. In *High Performance Distributed Computing*, 1999.
413. Virtual Organisation Membership Service (VOMS). <http://hep-project-grid-scg.web.cern.ch/hep-project-grid-scg/voms.html>.
414. G. von Laszewski. An Interactive Parallel Programming Environment Applied in Atmospheric Science. In G.-R. Hoffman and N. Kreitz, editors, *Making Its Mark, Proceedings of the 6th Workshop on the Use of Parallel Processors in Meteorology*, pages 311–325, Reading, UK, 2-6 Dec. 1996. European Centre for Medium Weather Forecast, World Scientific.
415. G. von Laszewski. Java CoG Kit Workflow Concepts. *accepted for publication in Journal of Grid Computing*, 2006.
416. G. von Laszewski. The Grid-Idea and Its Evolution. *Information Technology*., accepted for publication. Argonne National Laboratory, Argonne, IL 60439, U.S.A.
417. G. von Laszewski, K. Amin, S. Hampton, and S. Nijsure. GridAnt – White Paper. Technical report, Argonne National Laboratory, 31 July 2002.
418. G. von Laszewski, S. Fitzgerald, I. Foster, C. Kesselman, W. Smith, and S. Tuecke. A Directory Service for Configuring High-Performance Distributed Computations. In *Proceedings of the 6th IEEE Symposium on High-Performance Distributed Computing*, pages 365–375, Portland, OR, 5-8 Aug. 1997.
419. G. von Laszewski, I. Foster, J. Gawor, W. Smith, and S. Tuecke. CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids. In *ACM Java Grande 2000 Conference*, pages 97–106, San Francisco, CA, 3-5 June 2000.
420. G. von Laszewski and M. Hategan. Grid Workflow - An Integrated Approach. Draft Paper, 2005.

421. G. von Laszewski and D. Kodeboyina. A Repository Service for Grid Workflow Components. In *International Conference on Autonomic and Autonomous Systems International Conference on Networking and Services*. IEEE, 23-28 Oct. 2005.
422. G. von Laszewski, K. Mahinthakumar, R. Ranjithan, D. Brill, J. Uber, K. Harrison, S. Sreepathi, and E. Zechman. An Adaptive Cyberinfrastructure for Threat Management in Urban Water Distribution Systems. Technical report, Argonne National Laboratory, Jan. 2006. To be submitted.
423. G. von Laszewski, M.-H. Su, J. A. Insley, I. Foster, J. Bresnahan, C. Kesselman, M. Thiebaut, M. L. Rivers, S. Wang, B. Tieman, and I. McNulty. Real-Time Analysis, Visualization, and Steering of Microtomography Experiments at Photon Sources. In *Ninth SIAM Conference on Parallel Processing for Scientific Computing*, San Antonio, TX, 22-24 Mar. 1999.
424. G. von Laszewski, T. Trieu, P. Zimny, and D. Angulo. The Java CoG Kit Experiment Manager. Technical report, Argonne National Laboratory, June 2005.
425. W3C. Semantic Web Activity Statement. <http://www.w3.org/2001/sw/Activity>.
426. W3C. Web Services Addressing (WS-Addressing). <http://www.w3.org/2002/ws/addr/>.
427. W3C. Web Services Description Language (WSDL) Version 2.0 Part 2: Adjuncts. <http://www.w3.org/TR/2005/WD-wsd120-adjuncts-20050803>.
428. W3C. Web Services Choreography Description Language (WS-CDL) Version 1.0. <http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/>, 2004.
429. P. A. Walker and K. D. Cocks. HABITAT: A procedure for modelling a disjoint environmental envelope for a plant or animal species. *Global Ecology and Biogeography Letters*, 1:108–118, 1991.
430. I. Wang. P2PS (Peer-to-Peer Simplified). In *Proceedings of 13th Annual Mardi Gras Conference - Frontiers of Grid Applications and Technologies*, pages 54–59. Louisiana State University, February 2005.
431. I. Wang, I. Taylor, M. Shields, and A. Harrison. gridMonSteer: Generic Architecture for Monitoring and Steering Legacy Applications in Grid Environments. In *20th IEEE International Parallel and Distributed Processing Symposium. (Submitted in Consideration)*, 2005.
432. L. Wang, J.-J. Riethoven, and A. Robinson. XEMBL: distributing EMBL data in XML format. *Bioinformatics*, 18(8):1147–1148, 2002.
433. G. Wasson and M. Humphrey. Exploiting WSRF and WSRF.NET for Remote Job Execution in Grid Environments. In *19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*, 2005.
434. Web Tools Project. Eclipse web tools platform project. See web site at: <http://www.eclipse.org/webtools/>.
435. V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. Security for grid services. In *Twelfth International Symposium on High Performance Distributed Computing (HPDC-12)*, 2003.
436. Wikipedia. Choreography — wikipedia, the free encyclopedia, 2006. <http://en.wikipedia.org/w/index.php?title=Choreography&oldid=33366853> [Online; accessed 18-January-2006].

437. Wikipedia. Orchestration — wikipedia, the free encyclopedia, 2006. <http://en.wikipedia.org/w/index.php?title=Orchestration&oldid=34882858> [Online; accessed 18-January-2006].
438. Wikipedia. Petri net — wikipedia, the free encyclopedia, 2006. [http://en.wikipedia.org/w/index.php?title=Petri\\_net&oldid=35563598](http://en.wikipedia.org/w/index.php?title=Petri_net&oldid=35563598) [Online; accessed 18-January-2006].
439. E. O. Wiley, K. M. McNyset, A. T. Peterson, C. R. Robins, and A. M. Stewart. Niche modeling and geographic range predictions in the marine environment using a machine-learning algorithm. *Oceanography*, 16:120–127, 2003.
440. M. D. Wilkinson, D. Gessler, A. Farmer, and L. Stein. The BioMOBY Project Explores Open-Source, Simple, Extensible Protocols for Enabling Biological Database Interoperability. *Proc Virt Conf Genom and Bioinf*, 3:16–26, 2003.
441. B. Willke et al. The geo 600 gravitational wave detector. *Class. Quant. Grav.*, 19:1377–1387, 2002.
442. D. Willock, S. Price, M. Leslie, and C. Catlow. The relaxation of molecular crystal structures using a distributed multipole electrostatic model. *Journal of Computational Chemistry*, 1995.
443. K. Wolstencroft, T. Oinn, C. Goble, J. Ferris, C. Wroe, P. Lord, K. Glover, and R. Stevens. Panoply of Utilities in Taverna. In *UK e-Science All Hands Meeting*, 2005.
444. S. Woodman, S. Parastatidis, and J. Webber. Sequencing Constraints SSDL Protocol Framework. Technical Report CS-TR-903, University of Newcastle, 2005.
445. R. P. Woods. Automated Image Registration. <http://bishopw.loni.ucla.edu/AIR5/>.
446. W. Woods. What’s in a link: Foundations for semantic networks. In D. Bobrow and A. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*. New York: Academic Press, 1975.
447. A. Woolf, R. Cramer, M. Gutierrez, K. van Dam, S. Kondapalli, S. Latham, B. Lawrence, R. Lowry, and K. O’Neill. Semantic Integration of File-based Data for Grid Services. In *Workshop on Semantic Infrastructure for Grid Computing Applications*, 2005.
448. Workflow Management Coalition. Terminology & Glossary. Technical report, WfMC, 1999. <http://www.wfmc.org/>.
449. C. Wroe, C. Goble, M. Greenwood, P. Lord, S. Miles, J. Papay, T. Payne, and L. Moreau. Automating experiments using semantic data on a bioinformatics grid. *IEEE Intelligent Systems*, 19(1):48–55, 2004.
450. C. Wroe, R. Stevens, C. Goble, A. Roberts, and M. Greenwood. A Suite of DAML+OIL Ontologies to Describe Bioinformatics Web Services and Data. *The International Journal of Cooperative Information Systems*, 12(2):597–624, 2003.
451. Web Services Description Language (WSDL) 1.1. Technical report, W3C, 2001.
452. www.oasis open.org. Web services for remote portlets. Web Page.
453. L. Young. Scheduling componentised applications on a computational grid. MPhil Transfer Report, 2004.
454. L. Young, A. McGough, S. Newhouse, and J. Darlington. Scheduling Architecture and Algorithms within the ICENI Grid Middleware. In *UK e-Science All Hands Meeting*, pages 5–12, Nottingham, UK, Sept. 2003. ISBN 1-904425-11-9.
455. J. Yu and R. Buyya. A novel architecture for realizing grid workflow using tuple spaces. In *GRID*, pages 119–128, 2004.



456. J. Yu and R. Buyya. A Taxonomy of Workflow Management Systems for Grid Computing. Technical Report GRIDS-TR-2005-1, Grid Computing and Distributed Systems Laboratory, University of Melbourne, March 2005.
457. J. Zhao, C. Wroe, C. Goble, R. Stevens, D. Quan, and M. Greenwood. Using Semantic Web Technologies for Representing e-Science Provenance . In *3rd International Semantic Web Conference (ISWC2004)*, volume Springer-Verlag LNCS 3298, pages 92–106, 2004.
458. L. Zhao, P. Chen, and T. Jordan. Strain green's tensors, reciprocity and their applications to seismic source and structure studies. *Bulletin of the Seismological Society of America*, 2006.
459. Y. Zhao, J. Dobson, I. Foster, L. Moreau, and M. Wilde. A notation and system for expressing and executing cleanly typed workflows on messy scientific data. *SIGMOD Record*, 34(3):37–43, 2005.