Accessing Biodiversity Resources in Computational Environments from Workflow Applications

J. S. Pahwa*, R. J. White*, A. C. Jones*, M. Burgess*, W. A. Gray*, N. J. Fiddian*,
T. Sutton†, P. Brewer†, C. Yesson†, N. Caithness†, A. Culham†, F. A. Bisby†,
M. Scoble‡, P. Williams‡ and S. Bhagwat‡
*Cardiff School of Computer Science, Cardiff University, UK
Email :{j.s.pahwa, r.j.white, andrew.c.jones, m.burgess, w.a.gray, n.j.fiddian}@cs.cardiff.ac.uk
†School of Plant Sciences, The University of Reading, UK
Email: {t.sutton, p.w.brewer, c.yesson, n.caithness, a.culham, f.a.bisby}@rdg.ac.uk
‡The Natural History Museum, London, UK
Email: { m.scoble, p.williams, s.bhagwat}@nhm.ac.uk

***Abstract*— In the Biodiversity World (BDW) project we have created a flexible and extensible Web Services-based Grid environment for biodiversity researchers to solve problems in biodiversity and analyse biodiversity patterns. In this environment, heterogeneous and globally distributed biodiversity-related resources such as data sets and analytical tools are made available to be accessed and assembled by users into workflows to perform complex scientific experiments. One such experiment is bioclimatic modelling of the geographical distribution of individual species using climate variables in order to predict past and future climate-related changes in species distribution. Data sources and analytical tools required for such analysis of species distribution are widely dispersed, available on heterogeneous platforms, present data in different formats and lack interoperability. The BDW system brings all these disparate units together so that the user can combine tools with little thought as to their availability, data formats and interoperability. The current Web Services-based Grid environment enables execution of the BDW workflow tasks in remote nodes but with a limited scope. The next step in the evolution of the BDW architecture is to enable workflow tasks to utilise computational resources available within and outside the BDW domain. We describe the present BDW architecture and its transition to a new framework which provides a distributed computational environment for mapping and executing workflows in addition to bringing together heterogeneous resources and analytical tools.**

I. Introduction

Many individual scientists and institutions working with biodiversity data create their own resources, often for a narrow range of uses. Interoperability among such resources can sometimes be challenging, as they were not originally designed to be used together as part of a larger system and often do not conform to any recognised standard. Similarly, analytical tools to perform specific tasks have often been developed as stand-alone executables operating on data in a defined format so that if facilities from more than one tool are required, a user will frequently have to resort to transferring data between tools manually. There is a lack of automated means of correlating biodiversity and ecosystem data from various sources for its analysis and use in models and statistical tools to derive useful biological results [1].

For example, suppose a scientist wishes to investigate where a particular species of plant or animal might be expected to occur, given estimated past or predicted future climatic conditions. To investigate this bioclimatic modelling problem requires access to species distribution data, to tools that can model the climate characterising the locations where the species is to be found, to data pertaining to the climate at the time of interest, and to map images onto which the predicted distribution can be projected for viewing. Many tools routinely used by scientists to perform such experiments are legacy tools which are not interoperable.

In addition to the biodiversity resources and tools the scientist also requires access to computational resources using which biodiversity experiments could be conducted. The duration of an experiment depends upon its nature, number of variables involved and the amount of biodiversity resources required. For example, performing species distribution experiments for a large number of species under different climatic scenarios is complex and requires access to a pool of computational resources in order to achieve an acceptably short experiment duration. The challenge with providing access to computational resources is that these computational environments are not readily available and therefore they need to be designed and implemented based on available computational nodes, biodiversity resources, researchers' needs and using middleware such as Condor [2]. We require an environment which brings together heterogeneous resources and analytical tools into a single system where a user can combine tools with little thought as to their availability, data formats and interoperability. The environment should also meet the computational needs of a biodiversity experiment with the possibility of utilising computational resources which are external to the BDW domain so that the workflows created by researchers can be allocated the required computational resources.

The Biodiversity World (BDW) system provides a framework for biodiversity problem solving by providing access to widely dispersed and disparate data sources and analytical tools. The BDW system is a biology-led project and is being actively used for biodiversity informatics research in three main exemplar study areas: (a) biodiversity richness analysis

and conservation evaluation, (b) bioclimatic modelling and global climate change, and (c) phylogenetic analysis and biogeography. In all three of these research areas existing data sources and analytical tools are widely dispersed, available on different platforms and present data in different formats.

The BDW system provides scientists with tools which allow ready access to resources originally designed for use in isolation and the ability to compose these resources into complex workflows. It enables chaining of data processing operations and provides flexibility both in the choice of the kinds of resources to be used and in the sequence of operations to be performed for conducting biodiversity experiments. The system is extensible so that new resources and tools can be added to it.

We have outlined the requirements of the BDW system and the way they have influenced the architecture as part of an earlier publication [3]. An important requirement is to bring about interoperability between a diverse set of local and remote legacy databases and applications for a researcher to use these resources in designing complex workflows in a Grid environment. The architecture of the current BDW system is presented in [4]. However in the present paper we review the architecture of the present system again in order to familiarise the reader with the important system components summarised as follows.

1) The BDW system uses the Triana [5] workflow management tool for composing and executing workflows.
2) A communications layer called the Biodiversity World Grid Interface (BGI) insulates Triana from Grid resources and provides a standard mechanism for invoking operations on BDW resources.
3) BDW datatypes enable encapsulation and representation of various types of information and data for use by the components of the system. They are used for transporting data between endpoints within the distributed components of the system.
4) Resource wrappers provide a standard mechanism for performing operations on heterogeneous resources and analytical tools.
5) The Metadata Repository (MDR) provides a range of operations, through which information about resources can be obtained by a BDW workflow.

The present BDW system brings together heterogeneous resources for a researcher to conduct biodiversity experiments. It serves the purpose of providing a standardised set of operations and interfaces for accessing heterogeneous resources and executing workflow components on remote nodes very well. However the existing architecture also presents a number of challenges when it comes to accessing computational resources for executing workflow tasks. The architecture does not provide workflow managers with sufficient control and flexibility for accessing desired computational resources. It also does not provide the functionality of distributing user jobs across several nodes. Additionally the present architecture requires certain client-side libraries to be incorporated in the workflow manager for executing workflow tasks remotely. We believe that these restrictions have to be addressed so that workflows could be composed and executed with a greater degree of control and flexibility by a knowledgeable user where the user is free to choose the workflow manager of his choice for accessing resources in the BDW Problem Solving Environment (PSE).

There is a difference between the BDW resources and computational resources. BDW resources are those which we provide to users in the BDW PSE for accessing and analysing biodiversity data. They include analytical tools and heterogeneous data either from local or remote data sources. Computational resources are hardware and software facilities such as compute clusters and Grid middleware using which high-end computational capabilities and high-throughput computing can be achieved [6].

The new BDW architecture which is presently being developed and tested extends the existing architecture, thereby providing the user who wants the ability to control the execution of a workflow with a greater degree of flexibility. In this paper we describe both the current architecture and the new architecture and highlight the areas where significant improvements have been made. As part of the new architecture we use the existing mechanism of invoking operations on remote resources via resource wrappers deployed as Web Services. Based on the same Web Services model, in the BDW PSE, we also introduce a new resource invocation model which enables utilisation of computational resources for distributing workload across available nodes. By combining these two approaches as part of a single architecture we believe that not only our application, but applications from other domains can also benefit from the architectural design when accessing resources from workflow environments.

In the new architecture we introduce the Ganglia [7] cluster monitoring system and middleware from the Condor project to meet the computational needs of workflow tasks. The resources are deployed in a secure environment provided by chroot - a utility in Unix systems and we use rsync [8] for secure data transfer.

The paper is organised as follows. Section II provides information on three exemplar study areas. Section III describes the architecture of the current BDW system. Section IV presents a bioclimatic modelling workflow enacted using the current BDW PSE for modelling the distribution of species of a plant family, Fabaceae (bean family). In section V we describe the new BDW architecture. Section VI briefly describes some of the existing projects in the area of biodiversity, bioinformatics and biology which provide access to diverse resources and tools in their environments. Section VII presents conclusions and further work.

II. THE THREE EXAMPLE SCENARIOS

The BDW system is being actively used for biodiversity research in three main exemplar study areas. The three areas not only fall within the team's area of expertise but are also

representative of types of experiments which are performed in the area of biodiversity informatics.

A. Biodiversity Richness Analysis and Conservation Evaluation

A key issue in biodiversity science and one that also contributes directly to international conservation policy is the analysis of biodiversity richness patterns for a particular taxon (typically a group of species) around the world. When performing biodiversity richness analysis using the BDW system, the first step for an investigator using the system is to have a name for the target taxon. Because of instabilities in the nomenclature of species, databases containing relevant biological data may index data associated with that taxon using a different name from the one supplied by the investigator. Taxonomic verification, often the first step in any analysis using the BDW system, involves the retrieval of an authoritative list of names and synonyms. In our case this is obtained from the gateway of the Species 2000 (http://www.sp2000.org/) project [9] so that records indexed under any of these names can be retrieved.

In the second stage of the workflow, a distribution data set of specimens or observations belonging to the target taxon is composed from a variety of sources around the world. The final stage is for the distribution data set to be mapped to the WorldMap system [10], a specialist biodiversity analysis package designed to assist in selecting priority areas for biodiversity conservation. The WorldMap system uses species distribution data to compute a wide range of diversity measures, which it displays on a species richness map, that can then be used for further analyses. The simplest form of analysis is to identify areas of high species richness. As part of research in this area we are currently using species of butterflies from a database of Canadian butterflies provided by the Canadian Biodiversity Information Facility (CBIF) [11]. Access to the database is provided in the BDW system via a resource wrapper.

B. Bioclimatic Modelling and Global Climate Change

The subject of global climate change and its effect on the biological world is of great importance. Although significant progress has been made in recent years towards understanding how man is affecting the world's climate system, much less is known about how these changes are likely to affect the distribution and diversity of plant and animal species. A rapidly developing area of biodiversity analysis is to model the envelope of climatic and ecological conditions under which a single species lives, deducing this from known features of the places where it is recorded. Such a bioclimatic model can be used to calculate a potentially wider set of areas where the species might occur, or predict its future distribution under changing climatic conditions and to project these onto a map of the world. This can be used to predict the responses of the species which may become endangered or conversely become a pest presenting a new or increased threat.

C. Phylogenetic Analysis and Biogeography

Phylogenetic analysis comprises a variety of methods between groups for discovering the evolutionary relationships between groups of organisms, and typically produces an evolutionary tree, or phylogeny, to describe these relationships. The BDW system is used for phylogenetic analyses of various taxonomic groups. A standard phylogenetic workflow includes: searching the EMBL [12] DNA sequence database for sequences; using this to produce a phylogeny using parsimony or maximum likelihood techniques [13]; and estimating the age of species and lineages within the tree using techniques of temporal calibration [14]. This phylogenetic workflow can be integrated with the bioclimatic modelling workflow of section II-B. This has led us to the study of bioclimatic models from an evolutionary perspective, in particular to study the impact of historical climate change on the evolution of Mediterranean plant groups. We have developed ancestral bioclimatic models for the lineages of the sundews (a group of carnivorous plants in the family Droseraceae), and the popular garden flower genus *Cyclamen* (Myrsinaceae). These models have been projected into estimates of historical climate scenarios for 8-10 million years ago, to produce plausible estimates of ancestral distributions in geographical space. We have found that they demonstrate clear phylogenetic patterns coincident with historical shifts in climate, and they are helping us to understand the impact of climate change on the evolution of plants.

III. THE CURRENT BDW SYSTEM ARCHITECTURE

From the perspective of a biodiversity researcher, the system consists of a graphically based stand-alone workflow tool. The system allows linking of entities representing the resources, as part of its PSE, to create a workflow. By linking entities the user brings together several local or remote resources, legacy applications and analytical tools for conducting biodiversity experiments. A typical experiment involves retrieval of data from local or remote data sources and its processing by one or several applications or tools in a sequence or in parallel to derive useful results. The BDW system makes available all these tools and resources as part of its PSE even though they were originally designed for use in isolation.

From a system perspective, the BDW system has a multilayered architecture where interoperable components of each layer provide a set of operations and abstract the functionality of components of lower level layers from the layers above. Adopting a multilayered architecture has enabled the usage of remote resources for executing workflow tasks. Fig. 1 illustrates the present BDW system architecture. We describe important components of the system below.

A. Triana Workflow Management System

The BDW system uses Triana to provide workflow capabilities. The reasons for choosing Triana include its portability with different systems using Java, easy workflow assembly and its window-based visualisation tools for both designing and executing workflows. It has embedded features for authoring

tasks and workflow units and provides tools for working with Web Service-based systems. A key point is that Triana has been developed locally at Cardiff and we have direct access to the Triana team for support. The version of Triana used in the BDW system has been extended to support resources and analytical tools for biological analyses in our Web Services-based Grid environment. In our extended version of Triana, the toolbox contains tools provided for use in the BDW system in addition to standard Triana tools. This includes tools pertaining to resource wrappers, helper tools for accessing remote resources, configuration tools, data parsing tools and datatypes. The tools can be simply dragged onto a workflow design panel as workflow units and linked with other workflow units to quickly assemble a workflow, as in the example shown later in Fig. 4. The tools are provided as part of BDW client-side libraries and integrated with the Triana workflow system.

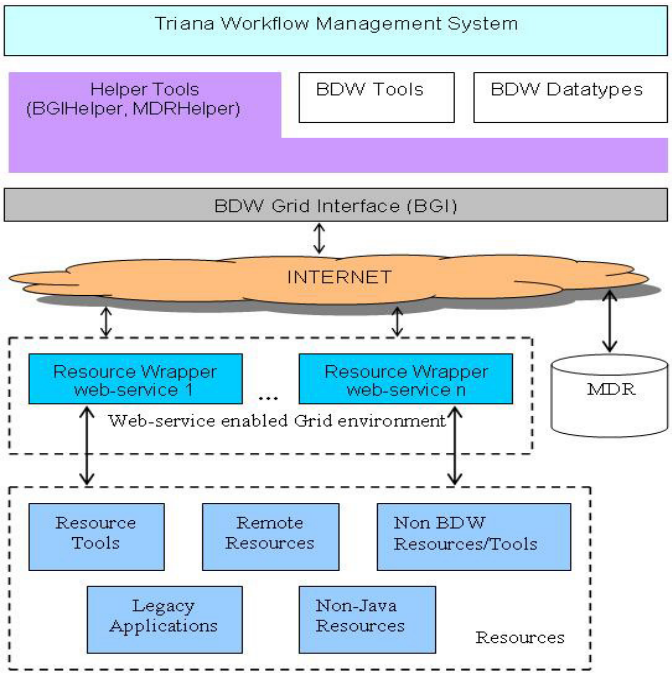

Fig. 1. Current BDW system Architecture

B. Resource Wrappers

In the BDW system, biodiversity resources are wrapped using the BDW resource wrappers which provide an invocation mechanism that allows any operation to be invoked in a standard manner using the same method call. A resource can be a remote resource such as a remote database or remote application, a legacy application within the BDW domain, a resource-accessing tool using a mechanism such as JDBC, and non-Java resources and tools written in C++ or other programming languages. Resource wrappers provide a standard operation called *invokeOperation*, with which dissimilar sets of operations supported by each resource can be invoked. The *invokeOperation* method allows invocation of operations defined on a resource inside its body. As the logic of the

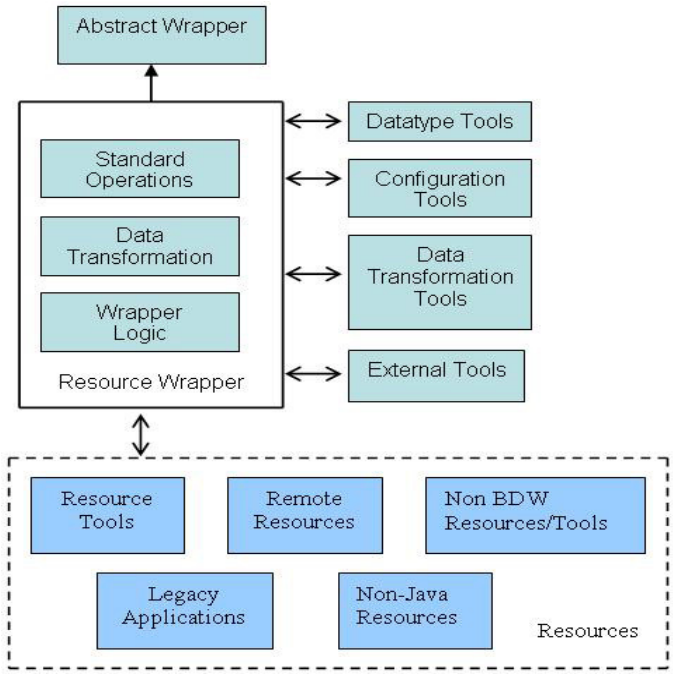

Fig. 2. Resource Wrapper Architecture

resource wrapper is implemented inside the *invokeOperation* method, different operations on a resource can be invoked by providing a different set of input values to the method which corresponds to the operation being invoked. Each resource wrapper is tailored for its resource but presents a standard interface to the BDW system components. The BDW resource wrapper architecture is presented in Fig. 2. The lower part of the diagram shows several types of entities which can be wrapped for the BDW system. A given resource wrapper wraps one such entity at a time. Resource wrappers use one or several different types of tools as illustrated in the figure when executing the wrapper logic. Some resource wrappers also do transformation of data so that data in an appropriate format can be passed as an input to the next unit in a workflow.

We have implemented wrappers for remote resources, local data stores or data stores accessible via JDBC drivers, cache databases, command-line tools, legacy applications such as WorldMap [10] and MATLAB, environmental modelling tools such as openModeller [15], wrappers for performing batch queries and wrappers for cataloguing, retrieval and visualisation of spatial data (including model outputs) within the BDW environment. At the time of writing, the BDW system provides researchers with 23 wrappers supporting a number of remote data resources from various parts of the world, local data sources, analytical tools and legacy applications. Some of the important remote resources wrapped in the BDW include the data portal of the Global Biodiversity Information Facility (GBIF) [16] and Australia's Virtual Herbarium (AVH) [17] - an online botanical information resource providing access to six million specimen records.

Resource wrappers implemented so far have been written

in the Java programming language and are deployed as Web Services using the Apache Axis [18] Web Services container. They can be accessed using the HTTP/XML-based SOAP messaging protocol. In Triana, a unit represents a wrapped resource or a tool. Triana invokes these wrapped resources as part of a workflow execution. Resource wrappers for resources are currently deployed at the Universities of Cardiff and Reading.

C. The BGI Layer

BDW workflow units access resource wrappers deployed in the Web Services environment via the BGI layer introduced in section 1 (see Fig. 1). The BGI layer provides standard communication objects for transporting data between a workflow unit and a resource wrapper. A communication object is an XML representation of a BDW datatype which enables encapsulation of resource-specific data for its transportation between the components of the system. The BGI enables invocation of the *invokeOperation* method on resource wrappers and presents the results to workflow units in the form of data communication objects called XmlDataCollection (XmlDC) and XmlRemoteData (XmlRD). Within the multi-layered architecture of the BDW system the BGI acts as a bridge between the workflow units and resource wrappers and insulates users from the complexities of resource wrappers deployed as Web Services. We provide a helper tool called BGIHelper for workflow units to use the BGI for invoking operations on resources. This provides workflow units with a high-level access to the *invokeOperation* method without needing to know about the underlying Web Services enabled Grid middleware which hosts resource wrappers.

D. BDW Datatypes

BDW datatypes are used by the components of the system for transporting data between end points via the BGI. For example, they are used between a BDW workflow unit inside Triana and a resource wrapper when an operation on a resource is invoked. They provide a single integrated mechanism for specifying different input parameters which a resource wrapper requires for performing operations on a resource and for returning the results. All the input parameters for a particular operation on a resource are specified as part of the attributes of a BDW datatype which can be transformed into a BDW communication object for delivering data between end points. Although more than one operation can be performed on a resource inside the *invokeOperation* method, the method itself is not overloaded. This is because a different set of parameters for each operation supported can be specified within the BDW datatype attributes. The resource wrapper, in its implementation of the *invokeOperation*, decides which operation to invoke on a resource based on the operation name specified by the user and its associated input parameter values in the BDW datatypes.

The data (which is usually in a format specific to the resource) returned from the resource wrapper after an operation is performed is also encapsulated within the BDW datatypes. By using this technique of encapsulating heterogeneous data within the standardised datatypes we hide the heterogeneity and enable system interoperability. The units in the BDW API provided in Triana toolboxes handle heterogeneous data. Data is either processed so that it can be represented to the user in a specified format (for example in a combo box) or it is used as an input to run the next unit in a workflow.

A datatype provides several attributes to hold different pieces of resource specific data. In some cases a datatype can encapsulate one or more sub-datatypes within itself. This design gives us the flexibility of transporting several pieces of unique data within a single datatype. Datatypes XmlDC and XmlRD are commonly used for transporting data objects between workflow units and resource wrappers. XmlRD consists of attributes for data such as a URI, filename, a boolean value and data itself represented as a character or byte sequence or as another embedded XML document. Fig. 3 describes a simple XML representation of the XmlDC datatype. The figure shows how more than one XmlRD datatype objects can be encapsulated within a single XmlDC.

```
<XmlDataCollection>
  <XmlRemoteData>
    <URI>http://www.bdworld.org</URI>
    <contents>input data</contents>
    <myBoolean>false</myBoolean>
    <filename>test.txt</filename>
  </XmlRemoteData>
  <XmlRemoteData>
    ...
  </XmlRemoteData>
</XmlDataCollection>
```

Fig. 3. XmlDataCollection Datatype

E. The Metadata Repository (MDR)

In order to locate resources and build workflows, metadata is needed to enable the selection of resources meeting appropriate criteria. Metadata is currently being used to provide information about resources available to the BDW system. The present interface to the MDR facilitates querying the repository to find essential information such as the operations supported by a resource, the parameters required by these operations and the returned data types, the location of the resource, and how it can be invoked. It is also possible to obtain status information for available resources, to ascertain which resources are online and available for use and which are not presently accessible.

Currently under development is an extension to the MDR whereby metadata is stored in an ontology. Represented in OWL (Web Ontology Language) this BDW ontology is accessed via an MDR interface developed using Protégé [19] and Jena [20]. This interface is available to the workflow environment, and other users, via a Metadata Agent Web Service. At present it holds the same information as the

previous MDR, but with the addition of semantic information allowing reasoning about available resources. For example, one of the semantic additions includes a facility for advanced port matching. Rather than suggesting compatible resources based on data type matching alone, the BDW ontology enables the inclusion of semantic information in the reasoning process.

IV. A BIOCLIMATIC MODELLING WORKFLOW

Currently the existing BDW system is being used to run bioclimatic modelling experiments on the whole of the plant family Fabaceae (bean family). This is a diverse family containing around 20,000 species representing around one twelfth of all flowering plants. They are distributed on all continents (excluding Antarctica) and are found within every major ecosystem. Global-scale models are currently being run under a number of climate scenarios and modelling algorithms in an attempt to quantify the potential risk of extinction of species of the family over the next 50 years. A degree of complexity is involved in managing data and tools from different sources and in different formats which the BDW system provides as part of its PSE. This research is modelling around 1800 species, using 3 bioclimatic modelling algorithms, under 4 climate scenarios consisting of 22 global climate surfaces: a total of approximately 22,000 bioclimatic modelling experiments.

Fig. 4 illustrates a complete bioclimatic modelling workflow built and run using Triana. The central tool *RunOpenModeller* is a tool based upon the OpenModeller [15] - an open-source species distribution modelling library providing a uniform method for modelling distribution patterns using different algorithms. The OpenModeller library is wrapped in a resource wrapper for use in the BDW environment.

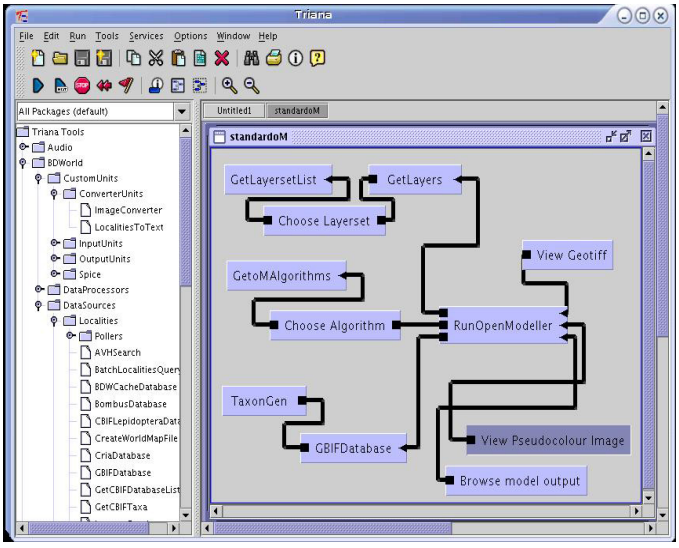

Fig. 4. A bioclimatic modelling workflow

RunOpenModeller requires three inputs:

1) The Localities data, which is provided by running the GBIFDatabase search tool, a resource wrapper for the GBIF portal [16]. This tool expects a string object pertaining to a taxon as an input and returns a localities collection (i.e. genus, species, latitude, longitude for each returned specimen) as output.
2) A Modelling algorithm such as GARP [21], BIOCLIM [22] or CSM [23] for modelling the potential distribution of species.
3) A collection of layers (typically climate layers such as temperature and precipitation values on a geographical grid) which are supplied by other workflow units.

Using the specified algorithm, OpenModeller constructs a bioclimatic model by interpolating the climatic data at the point localities of the specimen specified by the localities collection. The bioclimatic model for an area of interest is projected under present or predicted future climate parameters specified by an appropriate layers collection. It finally displays the results for interpretation by overlaying the projection onto an appropriate base map. Fig. 5 illustrates an example model output produced by executing the workflow illustrated in Fig. 4.

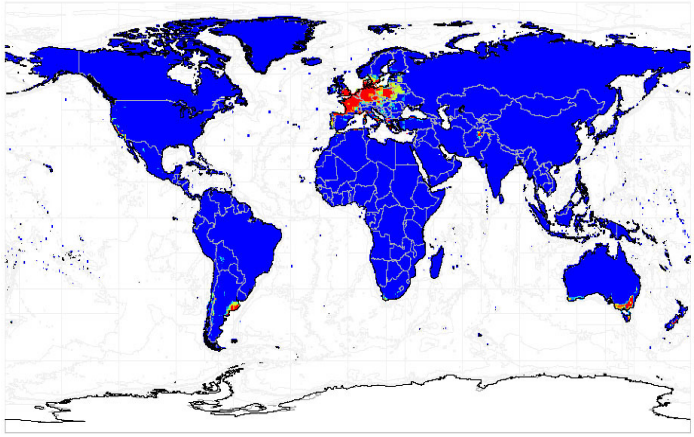

Fig. 5. Example model output for the clover species *Trifolium patens* Schreber (a member of the bean family). The map shows areas (shaded regions across Central and Eastern Europe, South America, Asia and Australia) predicted to be suitable for the species in the 2050's using the bioclimatic modelling algorithm GARP and the Hadley Centre climate model using the SRES A1F climate scenario [24].

V. THE NEW BDW SYSTEM ARCHITECTURE

The present BDW architecture effectively tackles the complexity of heterogeneous data and a diverse set of tools and brings them together as part of its PSE for a researcher to perform biodiversity experiments. However the present architecture does not fully meet the needs of the user in certain areas. We describe these areas below.

1) The current architecture which provides a Web Services access to biodiversity resources is dependent on certain client-side libraries. As described in section III-A, we have extended a version of Triana and incorporated BDW client-side libraries for invocation of remote resources and accessing tools in the BDW PSE. This approach works successfully but at the same time it

also restricts the use of other workflow managers for accessing the BDW PSE. It is not always possible to provide client-side library support for different workflow managers. Therefore what is required is the ability to use BDW resources without any client-side library support so that different workflow managers can access BDW resources and compose workflows in the BDW PSE.

2) Components of the present BDW architecture provide a uniform interface to heterogeneous resources via the *invokeOperation* method and encapsulate heterogeneous data in the standard BDW datatypes. This approach enabled interoperability of system components by virtue of identical interfaces. However it also restricted the visibility of operations available on resources and analytical tools to the user. It increased users' dependency on the MDR for gaining information about certain resources and tools before an operation could be invoked. This approach also required the availability of client-side libraries for creating standard operation calls and retrieving results data which is delivered to the workflow manager in encapsulated BDW datatypes. The new architecture resolves these issues and allows direct invocation of resources via their resource wrappers without the support of any client-side libraries. As part of the new architecture resource wrappers make operations available on resources more transparent and at the same time the issue of interoperability is also being addressed.
3) The present BDW architecture allows users to access a Web Service enabled Grid environment for invoking operations on remote resources and analysing biodiversity data. As part of the current architecture, resource wrappers are deployed at the Universities of Cardiff and Reading. A user can choose between the available sites for accessing BDW resources from the Triana running in the user's computer at the client-side. This approach is desirable because it provides users with a choice to use computational resource(s) from a preferred site. However at the same time it also limits the use of computational resources to only those nodes on which resource wrappers are deployed as Web Services. When possible we also desire the ability to use more than one node to perform parallel tasks and to achieve distribution of work load across several nodes when several users are trying to access BDW resources. We address these issues as part of the new architecture.

A. Approaches for Accessing BDW Resources

The new BDW architecture not only brings together heterogeneous resources and tools but also provides the ability to utilise available computational resources. This enhances the scalability of the system and provides the user with more control and choice in choosing appropriate computational nodes for executing workflow tasks. Fig. 6 illustrates the new BDW architecture which uses middleware from the Condor [2] project for distributing the workload across the available execute nodes. In the new architecture BDW resources can be

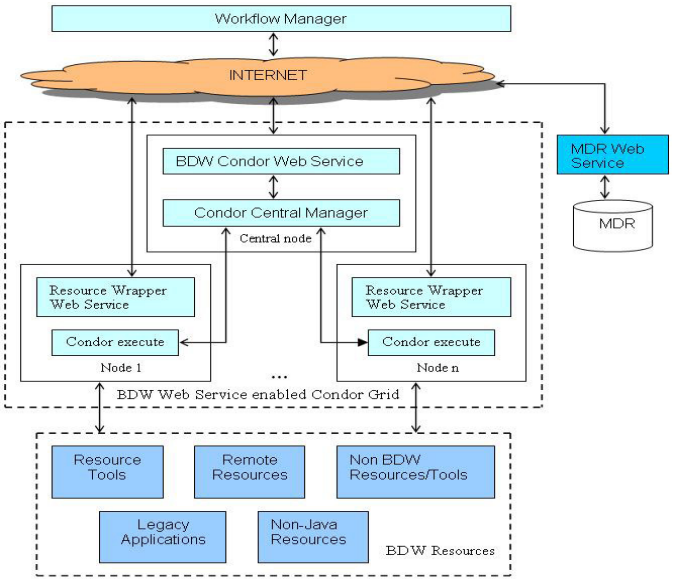

Fig. 6. New BDW system Architecture

accessed in the two different ways described below.

The first method of accessing resources in the computational pool is for advanced users who would like to first identify the resources which are available and then match the resources with the requirements of the workflow tasks. For example a user might be interested in nodes having any or all of the following capabilities such as higher processing speed, large disk storage, less network usage, lower average CPU load for a machine in a given duration of time, etc. Giving the user the ability to view host metrics provides users with the flexibility of using the node or nodes of their choice. In the BDW system, information about host metrics is provided by the Ganglia [7] cluster monitoring system. For example, Fig. 7 shows network statistics of one of the BDW cluster nodes. Once the user identifies the node, the workflow manager can directly access the resource wrapper Web Service running on the node. A workflow manager can be instructed to allocate a particular workflow task to the preferred node by creating a workflow unit which represents the task. Bindings between the workflow unit and the resource wrapper available on a preferred compute node can be established by importing the WSDL of the resource wrapper into the workflow environment. This design enables invocation of different resources available in different compute nodes when creating a workflow. However, this design requires replication of BDW resources across several nodes so that if a user's preferred node is not available or is busy serving another user, the user can choose from other available nodes which provide the same set of services. We provide users with tools for identifying nodes serving a particular resource via its resource wrapper. Matching of a workflow task with the preferred resource can also be achieved via automated means based on user requirements and a preferred host metric.

In the second method of accessing BDW resources available

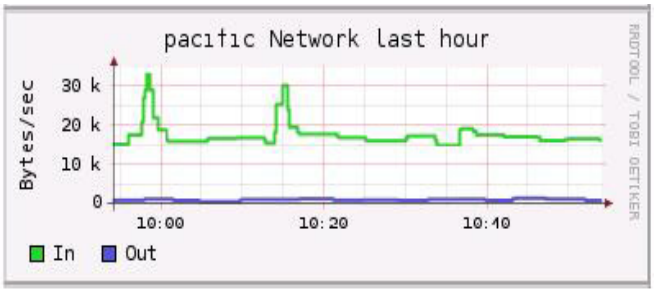

Fig. 7. Network statistics of a BDW Node

in the BDW computational environment a user submits a job to a BDW Condor web service component which creates a job description file and submits the job to the Condor central manager. The Condor central manager then decides which node to run the job on based on available nodes. Once the job execution is finished the user gets back results data or notification via the same process which invoked Web Service operation from the workflow environment. This design allows allocation of computational resources dynamically and running of a large number of iterative tasks perhaps with different variables every time and spreading of work load across available execute nodes. This design is also conducive for Grid environments where Grid resources such as computational nodes change dynamically. The Condor middleware provides a mechanism of first transferring the required libraries to the computational node before running a job. This will allow BDW tasks to be executed in dynamically changing computational environments. Condor also allows utilisation of computational resources outside the BDW domain via the mechanism of flocking. For example as part of a future possibility BDW jobs can be run in Condor pools available from wider university networks.

In both approaches, access to BDW resources available in the computational environment is provided directly via the Web Service interfaces. By combining the two approaches as part of the new architecture we have provided flexibility for workflow mangers to access computational resources of their choice. The two approaches can also be used together as part of a single workflow. While some tasks directly access a resource wrapper Web Service in a workflow, the others can be instructed to access resources via the BDW Condor Web Service. A node can be configured so that it is available to run jobs submitted by the Condor central manager; and also process resource wrapper Web Service requests which are submitted directly by workflow managers at the same time. This allows better utilisation of computational nodes and particularly those nodes with fast processors and better multitasking capabilities. We believe that the design decisions adopted in the new BDW architecture are also applicable in other application domains where workflow managers are required to access resources available in a Grid environment via Web Service interfaces.

The new BDW architecture makes the operations available on the BDW resources visible to the users and its invocation simpler without requiring client-side tools or the BGI layer so that different workflow managers can access resources of the BDW PSE. Resource operations are directly exposed as Web Services operations as opposed to embedding them inside the *invokeOperation* operation. As the work is still under development we are presently making available the resources required for running bioclimatic modelling workflow based on the new architecture progressively. The initial tests which involved creating and projecting species distribution models on to the map of the world using the *RunOpenModeller* tool in the Condor Grid environment has provided us with encouraging results.

B. Security and Data Transfer

In the BDW system, we are using chroot, a Unix system utility, to provide a secure environment for hosting BDW resources. The environment provides limited privileges for running hosted services without compromising overall system security. The chroot utility also provides a manageable solution for deployment of resources across computational nodes. By 'manageable' we mean rapid deployment, minimal configuration and easy replication of the setup from one node onto another. In the BDW system the need of large volumes of data transfer is less. One scenario where we use data transfer is when transferring model output files to a web server so that they can be accessed via a web browser. For transferring files between nodes and between sites we are currently testing the capabilities of rsync [8] - an open source utility that provides fast and incremental file transfer. The rsync utility uses a checksum-search algorithm for transferring just the difference between two sets of files across a network for efficient data transfer. The rsync traffic can be tunneled over the Secured Shell (SSH) protocol for secure file replication.

VI. RELATED WORK

The field of bioinformatics is diverse and a number of research projects in its different areas such as genetic studies, structural studies of cells and tissues, cellular processes, etc. have developed tools for the application of informatics techniques to biological data to address the challenges identified. The Web based myGrid project [25] provides middleware for conducting *in silico* experiments in biology. It provides access to a range of tools, services, information repositories and remote legacy bioinformatics applications which are wrapped as Web Services. It also adopts a workflow based approach by means of which a user can access the facilities provided by the environment when conducting experiments. One example application of the project is in the area of genetic studies for gaining new insights into diseases having a strong genetic component (such as Graves' disease) with the aim of aiding the process of designing novel therapies.

The GeneGrid [26] provides access and integration of disparate and heterogeneous applications and datasets from across the globe through the creation of a 'Virtual Bioinformatics Laboratory'. It provides access to resources and tools to

biologists interested in the development of antibodies and drugs. The BASIS project [27] serves the biology of ageing - at the cell, tissue and organism level by providing access to diverse biological resources to conduct experiments such as constructing a virtual ageing cell. The system provides tools such as SBML (Systems Biology Markup Language) which helps a researcher build a computer based model of ageing processes and test them. Some of the other projects in the area of bioinformatics such as e-Protein, BioSimGrid and e-HTPX are described in [28].

The goal of the Science Environment for Ecological Knowledge (SEEK) [29] System is to build a cyber infrastructure for gaining global access to ecological data and information and utilising distributed computational services for extending ecological and biodiversity analysis and research capabilities. The system provides data management capabilities, a semantic mediation system and a visual environment for working with biodiversity resources.

Pegasus portal [30] is a web based computational portal which allows access to Grid resources via a standard web browser using HTTP(S) protocol. This portal provides an approach for creating abstract workflows using Chimera [31] by specifying the metadata description of the desired data to be generated or analysis to be done. The portal supports two applications, LIGO [32] in gravitational-wave astronomy and Montage [33] for generating astronomical image mosaics. When executing a workflow using Pegasus Portal the mapping of tasks in the workflow to resources is done by the Pegasus System [34]. The Pegasus System maps abstract workflows to their concrete forms for their execution in the Grid environment. The abstract workflows can be constructed using Chimera which expresses workflow descriptions in the form of Chimera's Virtual Data Language (VDL). The concrete workflows produced by Pegasus are submitted to DAGMan [35] for execution. DAGMan is a meta-scheduler which allows submission of jobs to Condor in an order represented by a DAG (Directed Acyclic Graph) and processes the results.

The P-GRADE [36] Grid Portal is a workflow-oriented Grid portal which enables execution of job workflows using Globus technology based Grid middleware. By using the P-GRADE portal a user can create, execute and monitor workflows in Grid environments through high-level, graphical Web interfaces. The GENIUS [37] portal provides web-based access to the EGEE Grid infrastructure. It supports a number of Virtual Organisations. For accessing Grid resources and running jobs in the Grid environment, the GENIUS portal makes available to the members of VOs a set of services for submitting jobs, managing user files, proxies and data. It also provides interactive services for interactive analysis and services for monitoring Grid resources.

In the BDW system we have adopted an approach of first allowing the user to discover the resources and tools in the Grid environment which are accessible via the Web Service interfaces and then chaining them using a workflow manager to produce useful results. In the BDW PSE, although the execution of workflow tasks takes place in the Grid environment the workflow manager which invokes the resources is a desktop application. This design not only allows us to have a standard mechanism of working with heterogeneous BDW resources via Web Services but also provides us with the ability to combine the two approaches of working with BDW resources in a Grid environment in two different ways into a single system as described in Section V-A. Additionally, by using a Web Service based approach with a workflow manager it is possible for workflow tasks to interact with users whilst a workflow execution is in progress. User interaction aids workflow progress. For example a task (of a workflow in motion) might request user input, or request a user to select a value from a given set of values produced by a previous or current task which is then passed as an input to the next workflow task. A workflow task might also display intermediate results to the user indicating the progress that has been made so far and also allowing the user to make a judgment whether the progress achieved so far is desirable and expected.

VII. CONCLUSIONS AND FURTHER WORK

The BDW system brings together disparate resources and analytical tools for biodiversity researchers to solve problems in biodiversity and analyse biodiversity patterns. The system allows linking of these tools and resources into a workflow so that different activities performed as part of an experiment are automated and the experiment as a whole is conducted more efficiently. This is an improvement on the manual process of performing each activity individually using individual systems which are not linked. The new BDW architecture provides users with the flexibility to utilise computational resources either by submitting workflow tasks to the BDW Condor Web Service or invoking Web Service enabled resource wrappers directly whilst conducting an experiment in the BDW PSE. The BDW system, based on the present architecture is being actively applied in three exemplar study areas of biodiversity informatics. However we are also making available BDW resources as part of the new architecture progressively.

The BDW services in the new environment are hosted in a secure environment provided by chroot. In addition to a secure environment we also need a secure access to the BDW resources. A secure access to BDW resources is important for providing data of endangered species or species under specialist conservation programs to its authorised users only. How to provide a secure framework for resource access is currently being considered. We are evaluating the capabilities of the Shibboleth [38] web-based federated security model which provides a secure mechanism for inter-institutional sharing of resources.

The BDW PSE is based on a service-oriented architecture for providing access to its resources. Analytical tools and heterogeneous resources available through the BDW PSE as Web Services lack in user presentation facilities. The only visual environment available to the BDW resources is via the GUI based workflow tools such as Triana. By using the technology of portals presentation capabilities can be provided

to software components [39]. As part of future work in this direction we aim to provide browser based access to BDW resources using portals for better visualisation of BDW resources and to further simplify the process of generating and executing workflows.

ACKNOWLEDGMENTS

The project is funded by a research grant from the UK Biotechnology and Biological Sciences Research Council (BBSRC). We are grateful to a good number of collaborators for making data and tools available to the project. In particular we thank Species 2000 and the Hadley Centre for Climate Prediction and Research for providing us with valuable resources. We also thank the Triana team for the support they provided to the project.

REFERENCES

[1] D. Maier, E. Landis, J. Cushing, A. Frondorf, and A. Silberschatz, "Research Directions in Biodiversity and Ecosystem Informatics," in *Report of an NSF, USGS, NASA Workshop on Biodiversity and Ecosystem Informatics*, J. L. Schnase, Ed., NASA Goddard Space Fight Center, Greenbelt, Maryland, June 22–23, 2001.

[2] D. Thain, T. Tannenbaum, and M. Livny, "Distributed Computing in Practice: The Condor Experience," in *Concurrency and Computation: Practice and Experience*, vol. 17, no. 2-4, 2005, pp. 323–356.

[3] 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 *Grid Computing in Life Science: First International Workshop on Life Science Grid, Revised selected and invited papers (LNCS/LNBI 3370)*. Kanazawa, Japan: Springer-Verlag, 2004.

[4] J. S. Pahwa, P. Brewer, T. Sutton, C. Yesson, M. Burgess, X. Xu, A. C. Jones, R. J. White, W. A. Gray, N. J. Fiddian, F. A. Bisby, A. Culham, N. Caithness, M. Scoble, P. Williams, and S. Bhagwat, "Biodiversity World: A Problem-Solving Environment for Analysing Biodiversity Patterns," in *6th IEEE International Symposium on Cluster Computing and the Grid (CCGRID 2006)*, Singapore, May 16–19, 2006.

[5] I. Taylor, M. Shields, I. Wang, and R. Philp, "Grid Enabling Applications using Triana," in *Workshop on Grid Applications and Programming Tools*, Seattle, USA, 2003. [Online]. Available: http://users.cs.cardiff.ac.uk/Ian.J.Taylor/CV/Papers/GAPT_2003.pdf

[6] I. Foster and C. Kesselman, "Computational Grids," in *The Grid: Blueprint for a New Computing Infrastructure (Chapter 2)*. Morgan-Kaufman, 1999. [Online]. Available: http://www.globus.org/alliance/publications/papers/chapter2.pdf

[7] (2006) The Ganglia Monitoring System. [Online]. Available: http://ganglia.sourceforge.net/

[8] (2006) rsync. [Online]. Available: http://www.samba.org/rsync/

[9] 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," in *Proc. 11th International Conference on Database and Expert Systems Applications*. London, UK: Springer-Verlag, 2000.

[10] P. Williams. (2006) Biodiversity and WorldMap. [Online]. Available: http://www.nhm.ac.uk/research-curation/projects/worldmap/index.html

[11] (2003) The Canadian Biodiversity Information Facility Portal. [Online]. Available: http://www.cbif.gc.ca/portal/digir-toc.php

[12] (2006) The European Bioinformatics Institute Website. [Online]. Available: http://www.ebi.ac.uk/

[13] D. L. Swofford, "PAUP*: Phylogenetic Analysis Using Parsimony (* and other methods), Version 4.0 b10." Sinauer Associates, Sunderland, Massachusetts, 2002.

[14] M. J. Sanderson, "r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock," *Bioinformatics*, vol. 19, pp. 301–302, 2003.

[15] (2006) The openModeller. [Online]. Available: http://openmodeller.sourceforge.net/

[16] (2006) The GBIF portal. [Online]. Available: http://www.gbif.org/

[17] (2006) The Australia's Virtual Herbarium Website. [Online]. Available: http://www.chah.gov.au/avh/avh.html

[18] (2005) The Apache Web Services Project. [Online]. Available: http://ws.apache.org/axis/

[19] J. H. Gennari, M. A. Musen, R. W. Fergerson, W. E. Grosso, M. Crubézy, H. Eriksson, N. F. Noy, and S. W. Tu, "The evolution of Protégé: An environment for knowledge-based systems development," *International Journal of Human-Computer Studies*, vol. 58, no. 1, pp. 89–123, 2003.

[20] B. McBride, "Jena: A Semantic Web Toolkit," *IEEE Internet Computing*, vol. 6, no. 6, pp. 55–59, 2002.

[21] D. Stockwell and D. Peters, "The GARP modelling system: Problems and solutions to automated spatial prediction," *International Journal of Information Science*, vol. 13, pp. 143–158, 1999.

[22] H. A. Nix, "A biogeographic analysis of Australian elapid snakes," in *Australian Flora and Fauna Series Number 7*, R. Longmore, Ed. Canberra: Australian Government Publishing Service, 1986, pp. 4–15.

[23] M. P. Robertson, N. Caithness, and M. H. Villet, "A PCA-based modelling technique for predicting environmental suitability for organisms from presence records," *Diversity and Distributions*, vol. 7, pp. 15–27, 2001.

[24] C. Gordon, C. Cooper, C. A. Senior, H. Banks, J. M. Gregory, T. C. Johns, J. F. Mitchell, and R. A. Wood, "The Simulation of SST, Sea Ice Extents and Ocean Heat Transports in aversion of the Hadley Centre Coupled Model without Flux Adjustments," *Climate Dynamics*, vol. 16, pp. 147–168, 2000.

[25] C. Goble, C. Wroe, and R. Stevens, "The myGrid project: services, architecture and demonstrator," in *Proc. UK e-Science All Hands Meeting 2003 (AHM'03)*, Nottingham, UK, Sept. 2003.

[26] P. V. Jithesh, N. Kelly, S. Wasnik, P. Donachy, T. Harmer, R. Perrott, M. McCurley, M. Townsley, J. Johnston, and S. McKee, "Bioinformatics Application Integration in GeneGrid," in *Proc. UK e-Science All Hands Meeting 2005 (AHM'05)*, Nottingham, UK, Sept. 2005.

[27] T. B. L. Kirkwood, R. J. Boys, C. S. Gillespie, C. J. Proctor, D. P. Shanley, and D. J. Wilkinson, "Towards an E-Biology of Ageing: Integrating Theory and Data," *Nature Reviews Molecular Cell Biology*, vol. 4, pp. 243–249, 2003.

[28] W. A. Gray and C. Thompson, "Bioinformatics and eScience," in *Proc. UK e-Science All Hands Meeting 2003 (AHM'03)*, Nottingham, UK, Sept. 2003.

[29] (2006) The SEEK Project Proposal. [Online]. Available: http://seek.ecoinformatics.org/Wiki.jsp?page=SEEKProjectProposal

[30] G. Singh, E. Deelman, G. Mehta, K. Vahi, M.-H. Su, G. B. Berriman, J. Good, J. C. Jacob, D. S. Katz, A. Lazzarini, K. Blackburn, and S. Koranda, "The Pegasus Portal: Web Based Grid Computing," in *Proc. ACM symposium on Applied computing (2005)*, Santa Fe, New Mexico, 2005, pp. 680–686.

[31] I. Foster, J. Voeckler, M. Wilde, and Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation," in *Proc. 14th Conference on Scientific and Statistical Database Management*, 2002.

[32] 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*, vol. 256, no. 5055, pp. 325–333, 1992.

[33] G. B. Berriman, J. C. Good, A. C. Laity, A. Bergou, J. Jacob, D. S. Katz, E. Deelman, C. Kesselman, G. Singh, M. Su, and R. Williams, "Montage: A Grid Enabled Image Mosaic Service for the National Virtual Observatory," vol. 314, 2003.

[34] 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," 2004.

[35] (2006) Directed Acyclic Graph Manager. [Online]. Available: http://www.cs.wisc.edu/condor/dagman/

[36] (2005) P-GRADE Grid Portal. [Online]. Available: http://www.lpds.sztaki.hu/pgportal/

[37] (2006) GENIUS Portal Overview. [Online]. Available: http://egee.cesnet.cz/en/user/genius-guide.pdf

[38] (2006) The Shibboleth Project - Internet2 Middleware. [Online]. Available: http://shibboleth.internet2.edu/

[39] M. Osmond and Y. Guo, "Adopting and Extending Portlet Technologies for e-Science Workflow Deployment," in *Proc. UK e-Science All Hands Meeting 2005 (AHM'05)*, Nottingham, UK, Sept. 2005.