

# Data Grid Services for Biodiversity Informatics

Lester K.W. Leong, Paul Coddington, Andrew Wendelborn

School of Computer Science, University of Adelaide  
Adelaide, SA 5005, Australia

{lester,paul,andrew}@cs.adelaide.edu.au

## Abstract

The infrastructure provided by Grid computing, in particular Data Grids, allows a systematic approach to the process of discovering, accessing, analysing and integrating huge amounts of data. Most data grid software and applications assume the data is in files, although recently tools such as OGSA-DAI have been developed to provide grid services for interfacing to databases. This paper investigates the use of Data Grid technologies (Web Services, Grid Services and Grid tool) for applications that store all their data in databases, in particular, applications in biodiversity informatics. The aim of this work is to analyse the benefit (or otherwise) of these approaches. Web Service and Grid Service prototypes of the Global Biodiversity Information Facility (GBIF), a distributed web-based biodiversity information system, were developed as part of the evaluation process.

**Keyword:** Data grid, web service, grid Service, biodiversity informatics

## I. Introduction

The use of Grid Computing to harness substantial resources, including data and storage, compute cycles and computer-controlled experimental equipment, and allow them to be shared in a distributed environment, has enabled large complex problems to be solved more easily and faster than before. This is made possible because the Grid uses standardized protocols that allow geographically distributed, heterogeneous computation platforms to be linked together creating an illusion of a large-scale computer. Hence, over the past decade, several projects in science, engineering and even the business sector have turned to the use of Grid Computing. Data-Intensive Grid Computing, or Data Grids, are terms used for the part of the Grid infrastructure that allows a systematic approach to the process of discovering, accessing, processing and integrating large amounts of data.

The main focus of Grid computing middleware is on file-based storage [1], where data generated from scientific experiments, such as high-energy physics or astronomy, are stored in large files for further research or analysis. The total data sizes for these applications are measured in terabytes or petabytes. However, there are many other scientific applications, such as biodiversity informatics, that store data in databases rather than flat files. These applications are also data intensive, with millions (and potentially billions) of records stored in hundreds to thousands of distributed databases. The Global Biodiversity Information Facility (GBIF) [8] is an example of a biodiversity informatics

application that currently integrates hundreds of databases around the World to provide over 78 million records of biological specimens. More recently, initiatives such as the Open Grid Services Architecture Data Access and Integration (OGSA-DAI) project [10] have been working to provide grid services for accessing and querying data (or metadata) from databases, in order to support these kinds of applications.

The aim of this paper is to investigate how Data Grid technologies and services may be applied to the field of biodiversity informatics, and whether these methodologies are indeed beneficial to such applications. Hence, a Web Service (WS-GBIF) and a Grid Service (GS-GBIF) prototype of GBIF were implemented using the Service-Oriented Architecture (SOA) approach, which aims to provide a flexible and scalable loosely-coupled system that allows both human-to-application and application-to-application interactions. This approach is in contrast to the traditional web applications that use server-based technologies such as Common Gateway Interface (CGI).

The current implementation of GBIF uses a standard CGI-based web implementation, however, the developers are aiming to move to a SOA approach based on Web Services, and have proposed an outline for this architecture [2]. This was used as a starting point for the development of a Web Service (WS-GBIF) and a Grid Service (GS-GBIF) implementation of GBIF, with certain aspects simplified or omitted in the interest of achieving an experimental prototype. The GBIF developers are also working on developing a prototype Web Services implementation [3], although currently it is not a true Web Service implementation in that it still uses CGI to pass XML-encoded information. This is different from WS-GBIF, which uses Sun J2EE's Java Web Services Developer Package (JWSDP 1.3) [9], and GS-GBIF, which uses the OGSA-DAI toolkit. In addition, we have also developed useful features like distributed querying and caching of query results, that have yet to be implemented by GBIF.

This paper will first briefly discuss the area of concerns with biodiversity informatics, followed by a general introduction on the components and functions of GBIF. Subsequently, the development details of the WS-GBIF and GS-GBIF prototypes are presented before concluding with an evaluation of the different approaches.

## **II. Biodiversity Informatics**

Biodiversity data encompasses the variety of all life forms, spanning different plants, animals and micro-organisms, their genes and the ecosystem they exist within [11]. Biodiversity informatics is the use of Information Technology to organize, manage and analyse biodiversity data from various collections and experiments, in order to allow a more coordinated and systematic approach to the sharing of biodiversity information.

Unfortunately, most biodiversity information (like most information in many other fields of science) is not readily available online for use by researchers [4]. It is estimated that there are over two billion biological specimens stored in natural history collections world-wide. The information describing most of these specimens is only available in paper form. Many museums and herbaria are now in the process of capturing their specimen information in electronic form and storing it in a database. However each institution will typically have a different database schema to describe this data, making it very difficult to do detailed analysis of specimen information across collections from multiple organisations.

To enable users to access data online in a uniform fashion, standardised descriptive data, known as metadata, has to be used to properly describe the data [5]. The metadata is usually stored along with the scientific data it describes in the same file, or else in a separate file or database. In some cases, the distinction between what is metadata and what is data is rather subjective [4]. In the case of biological specimen data, all of the information about the specimen is stored together in a database, and a standard schema for describing all of this data and/or metadata is required. This information includes taxonomic data, such as the specimen's scientific name, vernacular (common) name, synonyms (other scientific names that the taxon is or was known by) and taxonomic hierarchy, and detailed information about when and where the specimen was collected or sighted.

There are four schemas that are commonly used in biodiversity informatics to facilitate distributed or federated queries across numerous databases. Dublin Core [12] is a domain-independent metadata schema with 15 fields that provide generic information on a particular data resource, such as a database, for resource discovery. Darwin Core [13] is a domain-specific schema that extends Dublin Core, and is aimed at natural history specimen collections and observation databases. The approximately 44 fields in this schema describe generic (i.e. independent of the kind of organism) information about the time and location of the observation record or the collection of the specimen, along with information about the collection holding the record, such as zoological, botanical and genetic resource collections. Herbarium Information Standards and Protocols for Interchange of Data (HISPID) [14] is a sub-domain specific metadata schema, which provides details specific to botanical collections. It has approximately 145 fields. The Access to Biological Collections Databases (ABCD) [15] schema, unlike the above-mentioned schemas, adopts a hierarchical format that is deeply nested to allow the schema to be extensible for different sub-domain specific collections. Hence, it uses approximately 150 of its over 300 fields to describe the specifics of the sub-domains.

Australia's Virtual Herbarium (AVH) [16] was the first large-scale biodiversity informatics project to federate many distributed databases of specimen information. It is a collaboration between all nine State, Commonwealth and Territory herbaria in Australia, using the HISPID schema to make all their botanical specimen information accessible via a web interface. On a larger scale, GBIF is an international collaboration that aims to provide federated access to all of the World's primary biodiversity data, by linking thousands of databases around the world using standard schema to describe the data, and standard interfaces to index, query and access the data.

### **III. Global Biodiversity Information Facility (GBIF)**

GBIF is an enormous international effort with currently (June 2005) 130 data providers offering over 78 million records of biological specimen information from 541 databases distributed all over the World. GBIF's extensive membership and collection of specimen records is still growing, at approximately 2 million specimen records a month. It is the world's largest biodiversity information system, and one of the largest distributed scientific databases in the world, hence the motivation for choosing it as an example for the work presented in this paper. GBIF currently provides two mechanisms for federating databases into the system [2]. It uses the Distributed Generic Information Retrieval (DiGIR) [17] protocol for performing single queries to retrieve structured data described in Darwin Core format from distributed heterogeneous databases. Alternatively, Biological Collections Access Service (BioCASE) [18], which is an equivalent to DiGIR, will be used for databases supporting the ABCD schema. Using DiGIR or BioCASE, GBIF extracts the main fields of interest from all the specimen records of all the data providers, and generates a summary index, which is stored in a central database in the GBIF Portal.

The Proposed GBIF network consists of 3 core components:

- a. GBIF Portal* - The GBIF Portal is a unique node on the GBIF network that acts as a gateway to the aggregated data services it provides and also the different Web Services offered by all the nodes. Therefore, a query made on GBIF Portal will allow the user to obtain a list of indexed specimen summary data of the queried species, which gives a much faster response than doing a distributed query over hundreds of databases. It also provides the user with an interface to the Web Services that are offered by the biodiversity data providers (Data Nodes) to perform extraction of detailed specimen records.
- b. Participant Node* - The Participant Node's main responsibilities are, to promote the inclusion of new biodiversity data providers, coordinate their registration process and to provide national language support if needed.
- c. Data Nodes* - The primary data service providers of GBIF, which offers biodiversity data that are shared freely within the GBIF network through Web Services.

In the WS-GBIF and GS-GBIF prototype implementations, only the GBIF Portal and the Data Nodes were developed because the Participant Node's functions are outside the scope of this paper.

The three core services proposed to be offered by the services-oriented version of GBIF and implemented in WS-GBIF and GS-GBIF are:

1. *Metadata Service* - This service provides access to the Service Registry, which is used to locate all the Data Nodes within the GBIF network.
2. *Specimen/Observation Service* - This service will be used to access metadata describing the specimens and observations of a particular species. The ABCD schema will be the primary schema used within GBIF, while Darwin Core will only be used as an interim schema until ABCD schema is more widely accepted and used.
3. *Taxonomic Name Service* - This proposed service offers the access to authoritative sources of data relating to taxonomic names such as the synonyms and vernacular names of the species, the list of child taxa and the higher taxonomic hierarchical ranks of the supplied scientific name.

#### **IV. Web Service Prototype of GBIF (WS-GBIF)**

This section discusses some of the architecture and component issues of WS-GBIF in relation to the proposed GBIF services-oriented architecture. The WS-GBIF prototype was developed using JWSDP tools, such as Java APIs for XML-Processing (JAXP), Java APIs for XML Registries (JAXR) and Java APIs for XML-based Remote Procedure Calls (JAX-RPC). The prototype follows closely to the generic Web Service architecture, where there are Service Requestor, Service Registry and Service Providers interacting with one another. Therefore, this particular architecture (illustrated in Figure 1) makes WS-GBIF generic enough for use in other applications that integrate federated databases.

Clients can query the summarised specimen records through the Portal Service's Data Index for quicker responses. Full specimen records may also be obtained or queried from the individual Data Nodes, provided that they are registered and published on WS-GBIF's registry server (Step 1 in Figure 1). A client can do a distributed query of the full specimen records (Step 2), the access points of the Data Nodes that offer the records will first be retrieved from the registry (Step 3) to perform the distributed queries (Step 4), and the results returned (Step 5) in either Darwin Core or ABCD Schema. The Portal Service then consolidates the results, stores them into the cache and subsequently formats them for display on the client's web browser (Step 6).

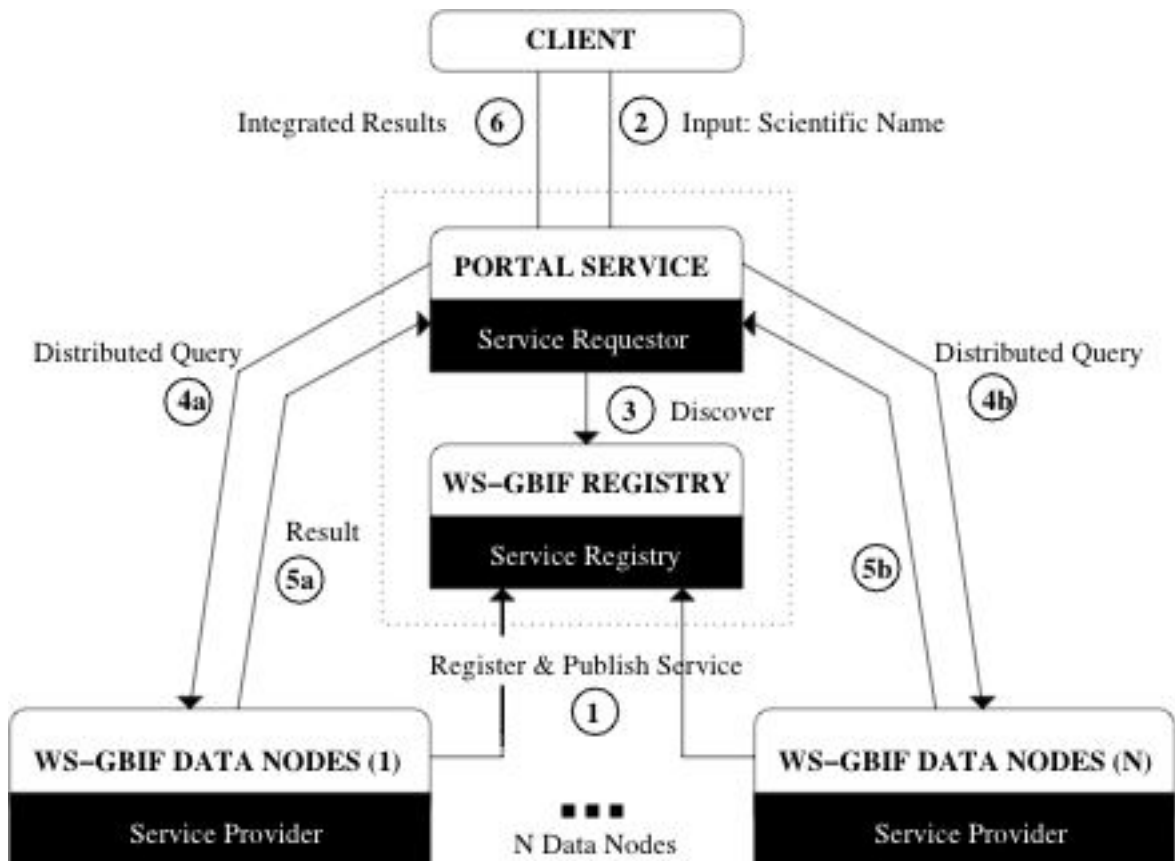


Figure 1: Web Service Prototype Architecture with Distributed Queries

#### A. *WS-GBIF Portal*

This section briefly describes components (based on the proposed GBIF services architecture) that constitute to the WS-GBIF Portal which coordinates and performs the user's request for biodiversity data offered from the Data Nodes.

**Registry Manager** is a core component within the GBIF Portal that consists of the registry that offers the Metadata Service. In WS-GBIF, this service is implemented using Java Registry Server and interfaced through JAXR, for the registration and publication of new nodes and services, or the modification and removal of existing nodes and services. On the other hand, the ongoing prototype development of the GBIF Portal uses Systinet's WASP implementation of a UDDI.

**Index Manager** - consists of a Biodiversity Data Index that contains a set of summary data for each record offered by the WS-GBIF network, including the name of the taxon to which the organism has been assigned; location and date at which the specimen was collected or the observation made; where the specimen is held and how to access more information. The purpose of accessing only the Data Index rather than all the Data Nodes is to provide quicker discovery of the specimen records that match the user's query. Moreover, the indexed data can serve as back-up if the Data Node's service becomes inaccessible. WS-GBIF and GS-GBIF implement the Data Index using MySQL.

**Taxonomic Name Service** provides the Search Engine with the taxonomic name information from the Electronic Catalogue of Names of Known Organisms (ECAT) [19], which is still under development. Hence, in WS-GBIF, a taxonomic database is implemented using MySQL in WS-GBIF to mimic the possible contents of the ECAT.

**Search Engine** makes use of the Index Manager to extract the list of summary records that matches the species search string. However, if the user is interested in retrieving the full record set, then the Registry Manager will be consulted to obtain the list of access points of Data Nodes that will take part in the distributed query. Using the access point, Java's JAX-RPC mechanism is used to dynamically invoke the remote Specimen/Observation Services of all involved Data Nodes, and the results are returned to the Portal Service. Any results in Darwin Core format are converted to ABCD format before presentation. Caching is also performed on the distributed query results and stored in a separate table within the local Data Index Database. Alternatively, the client may access the full record set from the cache table if the record is still in its active state or when the Data Node's service is unavailable.

**Data Connection Framework** handles all inbound connections made to the GBIF Portal, such as service requests and the return of detailed query results from a Data Node. In some cases, query results are transformed into the preferred metadata schema if they do not conform to the new or upgraded format used by GBIF. This transformation process allows the Data Nodes to delay the migration of their schema format and yet still be able to serve as data service providers within the network. WS-GBIF Portal uses JAXP to perform the parsing of the detailed specimen records from Darwin Core into ABCD schema.

**Presentation Service** provides the functionality to map XML data response documents into HTML for web display. While GBIF proposes the use of Extensible Stylesheet Language Transformations (XSLT) and Cascading Style Sheets (CSS), WS-GBIF uses JAXP to perform similar transformations and Java Server Pages (JSP) to display the results on the web browser.

**Session Manager** consists of several functions including caching result sets that were recently generated by a user-initiated detailed query. Other functionalities include logging of data accessed and Portal personalisation capability. In WS-GBIF, only the caching capability is implemented and it is handled by the Search Engine component.

## ***B. WS-GBIF Data Nodes***

The Data Node provides access to biodiversity data within the GBIF network through standard exchange interfaces, such as DiGIR and BioCASE. These data are usually stored on the Data Node's local databases.

In WS-GBIF's prototype, the service endpoints of the Data Nodes are implemented using JAX-RPC, together with JAXP to parse metadata schemas described in XML grammar. These Data Nodes do not have local databases but use a proxy to propagate the queries to existing biodiversity providers to obtain the required detailed specimen data. The purpose of using a proxy is to experiment on data exchanges using real biodiversity data and also as an interim measure that encourages biodiversity providers to offer their data on GBIF despite not being Web Services compliant. There are two categories of existing biodiversity providers and they are:

- a. **Non-GBIF Provider** - this refers to existing biodiversity providers that are currently not Data Nodes within the official GBIF network. The nine different nodes of the AVH represent this classification. The Data Node's proxy will redirect the SOAP request to the appropriate AVH node through a HTTP GET request. The results returned in HISPID format are parsed into ABCD schema and returned to the WS-GBIF Portal as a SOAP response document. Our implementation essentially provides a Web Service wrapper to the existing CGI-based query
- b. **Existing GBIF Provider** - this refers to existing biodiversity provider that are also registered Data Nodes within the GBIF network. The California Academy of Sciences (CAS) [20] represents this classification and supports the use of DiGIR retrieval protocol. Hence, the proxy service will run the DiGIR client application that connects to the DiGIR access point of CAS. The obtained results described in Darwin Core format are subsequently sent back to the WS-GBIF Portal as a SOAP response document.

## V. Grid Service Prototype of GBIF (GS-GBIF)

The purpose of developing a Grid Service prototype of GBIF, is to show how it would be possible for GBIF to move towards the use of Grid Services, and potentially take advantage of the Grid tools that are now becoming available for interfacing with distributed databases. Hence, GS-GBIF provides the framework needed to extract data from either the Data Index or perform distributed queries on the Data Nodes. OGSA-DAI was chosen as the Grid middleware to perform this functionality because it provides a customisable Grid Service interface to geographically distributed databases for data access and integration operations, without having to manage low-level complexities of JDBC connections. This work was done using release 4.0 of OGSA-DAI, which was implemented as OGSi grid services and worked with Globus Toolkit version 3.2.

The generic architecture adopted by GS-GBIF is similar to WS-GBIF, with the inclusion of Grid Service factories to provide lifetime and state information. However, unlike WS-GBIF, it does not use proxies for data retrieval from the Data Nodes, since the idea of OGSA-DAI is to provide grid services interfaces to databases. The use of OGSA-DAI to access the Data Index and the Data Node's databases allow these resources to be exposed on the Grid for other Grid applications to tap on the vast biodiversity data offered by GS-GBIF. This reflects the application-to-application form of interaction that is achievable through the SOA approach.

### A. *GS-GBIF Portal*

Similar to WS-GBIF Portal, the GS-GBIF Portal provides the core functionality to accept the user's search string and performs the necessary query on the Data Index for all the specimen summary records matching the name provided by the user. It also facilitates distributed queries to the Data Nodes if the user wishes to extract detailed specimen data from the

providers. However, other functionalities like schema transformations that have already been attempted in WS-GBIF will not be reflected in this prototype. Hence, the components that constitute to the GS-GBIF Portal are:

- a. **Portal Service** - provides the necessary functions to format the user request and consult the GS-GBIF Registry for the appropriate Grid Service, to either extract indexed summary data from the Data Index or detailed specimen data from the Data Nodes using OGSA-DAI. The extracted data are then formatted into web-browser readable form through the use of JSP.
- b. **GS-GBIF Registry** - supports the discovery, creation and destruction of the available Grid Data Service (GDS) interfaces to the different databases, through the use of a specialised registry for OGSA-DAI, known as the DAI Service Group Registry (DAISGR).
- c. **GS-GBIF Data Index** - Similar to the WS-GBIF's Data Index, the GS-GBIF Data Index holds an indexed summarised copy of all biodiversity records offered by the GS-GBIF network, which is accessed using OGSA-DAI.

Distributed queries are performed in parallel using Java threads. Each thread upon obtaining the handle of the Specimen/Observation GDS Factory creates an instance of the factory in order to retrieve detailed specimen records from the designated remote Data Node. It uses a pre-defined OGSA-DAI operation called `SqlQueryStatement Activity` that performs data access on the relational database of the Data Node, based on the SQL query statement formulated by the Portal.

### ***B. GS-GBIF Data Nodes***

Each Data Node is installed with OGSA-DAI and the pre-defined `SqlQueryStatement Activity`, which will be invoked upon receiving the GDS request. The prototype of GS-GBIF does not rely on any proxies or wrappers as in WS-GBIF to obtain detailed specimen data, because the key objective of using OGSA-DAI is to have it interface with a locally accessible database. The result is wrapped as a GDS response document and sent to the Portal for display on the web browser.

## **VI. Conclusion and Future Work**

The objective of this paper is to investigate how Web Services, Grid Services and Data Grid technologies may be applied in the field of biodiversity informatics and to address the concerns of providing an infrastructure for the efficient access of biodiversity data stored in multiple heterogeneous databases that are geographically distributed and independently managed.

Web Services provided a standard and straightforward way to develop a services-oriented implementation of GBIF, with all the usual advantages of the SOA approach. The WS-GBIF prototype that we have implemented allows a flexible and dynamic approach towards the binding of the user to the Data Nodes, without having prior knowledge of each other. Such an approach is necessary because the massive amount of shared biodiversity data will not be consolidated into a single physical location but geographically dispersed on numerous databases. Moreover, because the discovery of new species and the cataloging of their specimens is an on-going process, they must be loosely coupled to be integrated easily into the GBIF network. Although JWSDP provided several

useful tools, other middleware, such as Apache Axis, that can be used for the development of Web Service applications.

The issue of performance and scalability of various Web Service implementations needs to be investigated further, such as the effects of various SOAP encoding styles [7] and extensive XML parsing. This is because a distributed query can potentially extract thousands of records that are described in the highly descriptive ABCD schema.

While WS-GBIF was developed to show how Web Services could be applied to biodiversity informatics applications, GS-GBIF was not only to show that Grid Services could do the same job but also whether Grid tools, which were once focused on file-based data, could now work effectively on database-type applications like biodiversity informatics. However, most biodiversity informatics applications do not fully benefit from the fact that Grid Services are stateful and transient, since they are usually only focused on data access, like GBIF, and seldom involve a workflow of operations interlinked together. The standard Web Services approach is therefore perfectly adequate. Having said that, there may exist other more complex biodiversity data analysis that could benefit from stateful Grid Services to implement their functionalities. OGSA-DAI provided useful tools to easily generate a Grid Service interface to database access and querying. However for this particular application, as similar tool to provide a Web Service interface would have been more useful.

Caching and data replication are common techniques in distributed computing to increase performance in terms of quicker access times, data availability and fault tolerance. Such techniques apply to biodiversity informatics applications because many records may be queried or accessed from many widely distributed locations. AVH, which uses a distributed query mechanism, utilizes caching, however GBIF, which uses an indexing approach, currently does not. Our implementation of WS-GBIF implemented a simple and effective caching mechanism for distributed queries.

Biodiversity informatics applications, like GBIF, may wish to tap in the experience of the Grid community, by looking at the concepts and implementation details behind various Grid indexing tools such as Replica Location Index (RLI) and the Grid Index Information Service (GIIS) of the Globus Toolkit [21]. These may be useful for specimen indexing, although it is not clear that the algorithms used for indexing in these services would be scalable to such a large network of data providers. However the standard GIIS approach of using an index hierarchy would probably be better than the current GBIF architecture with an index for all providers. GBIF could be set up so that each data provider is indexed by a national portal, and then these are in turn indexed by multiple global GBIF portals. This would decentralize the indexing overhead, and also allow users to query national portals when they only want local information, thus reducing load on the main GBIF portals.

The Metadata Catalogue Service (MCS) [5][6], is also worth investigating since it can be customised using OGSA-DAI Activities to provide a mechanism for searching specimen information. Similarly the digital library community is developing web service standards for metadata querying, such as the Search and Retrieve Web service (SRW) [22], and standards such as the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [23] for indexing distributed databases. It may be better to leverage off the development of standard, general approaches to querying and indexing of distributed data rather than developing bespoke implementations from scratch.

Other areas for future work include how OGSA-Distributed Query Processing (OGSA-DQP) [24], which is an extension of OGSA-DAI, can be used in biodiversity informatics applications to perform optimised scheduling of workflows for distributed database accesses. Another key issue is the development of WS-RF [25], and whether the additional functionality of WS-RF can be of benefit to systems such as GBIF.

## References

- [1] M.N. Alpdemir et al., "OGSA-DQP: A Services-Based Distributed Query Processor for the Grid", In Proc. of UK e-Science All Hands Meeting, Nottingham, Sept 2003
- [2] D. Hobern, "GBIF Global Biodiversity Data Architecture", <http://www.gbif.org/prog/dadi/docu/>, Mar 2003.
- [3] GBIF Portal Web Services prototype, <http://www.gbif.net/>.
- [4] J.L.Schnase, "Research Directions in Biodiversity Informatics", In Proc. of the 26<sup>th</sup> International Conference on Very Large Databases, 2000.
- [5] G. Singh et al., "A Metadata Catalog Service for Data Intensive Applications", In Proc. Of Supercomputing 2003 (SC2003), Nov 2003.
- [6] E. Deelman et al., "Grid-Based Metadata Services", 16th International Conference on Scientific and Statistical Database Management (SSDBM04), Jun 2004.
- [7] F. Cohen, "Discover SOAP Encoding's Impact on Web Service Performance", <http://www-128.ibm.com/developerworks/webservices/library/ws-soapenc/>
- [8] The Global Biodiversity Information Facility, <http://www.gbif.org/>
- [9] Java Web Service Developers Package, <http://java.sun.com/webservices/jwsdp/>
- [10] Open Grid Services Architecture - Data Access and Integration, <http://www.ogsadai.org/>
- [11] Australian Government Department of Environment and Heritage, Biodiversity Website, <http://www.deh.gov.au/biodiversity>
- [12] Dublin Core Metadata Standard, <http://www.dublincore.org/>
- [13] Darwin Core, <http://darwincore.calacademy.org/>
- [14] Herbarium Information Standards and Protocols for Interchange of Data, <http://plantnet.rbg Syd.nsw.gov.au/HISCOM/HISPID/HISPID3/hispidright.html>
- [15] Access to Biological Collection Data Task Group, <http://www.bgbm.org/TDWG/CODATA/>
- [16] Australia's Virtual Herbarium, <http://www.chah.gov.au/avh/>
- [17] Distributed Generic Information Retrieval, <http://digir.sourceforge.net/>
- [18] A Biological Collection Access Service for Europe, <http://www.biocase.org/>
- [19] GBIF's Electronic Catalogue of Names of Known Organisms, <http://www.gbif.org/prog/ecat/>
- [20] California Academy of Sciences, <http://www.calacademy.org/>
- [21] The Globus Toolkit, <http://www.globus.org/>
- [22] Search and Retrieve Web Service (SRW), <http://www.loc.gov/z3950/agency/zing/srw/>
- [23] The Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), <http://www.openarchives.org/OAI/openarchivesprotocol.html>
- [24] Open Grid Services Architecture - Distributed Query Processing, <http://www.ogsadai.org/dqp/>
- [25] Web Service Resource Framework, <http://www.globus.org/wsrf/>



**Lester K.W. Leong** obtained his Computer Science honours degree from the University of Adelaide, Australia in 2004. He is currently working as a Research Officer for A-Star (Singapore Institute of High Performance Computing).



**Dr Andrew Wendelborn** is a Senior Lecturer in Computer Science at the University of Adelaide. His principal research interest is in grid computing, especially design and implementation of workflows, and middleware infrastructure for distributed high performance computing. Other interests include programming languages for parallel and distributed computing.



**Dr Paul Coddington** is a Senior Lecturer in Computer Science at the University of Adelaide and deputy director of the South Australian Partnership for Advanced Computing. His research interests are in high-performance computing, parallel algorithms, grid computing, data grids and computational science.