BioMOBY Successfully Integrates Distributed Heterogeneous Bioinformatics Web Services. The PlaNet Exemplar Case[1]

Mark Wilkinson*, Heiko Schoof, Rebecca Ernst, and Dirk Haase

James Hogg iCAPTURE Centre for Cardiovascular and Pulmonary Research, St. Paul's Hospital, Department of Medical Genetics, University of British Columbia, Vancouver, Canada (M.W.); Technische Universität München, Chair of Genome Oriented Bioinformatics, Center of Life and Food Science, D–85350 Freising-Weihenstephan, Germany (H.S.); and Institute for Bioinformatics, Gesellschaft für Strahlenforschung National Research Center for Environment and Health, D–85764 Neuherberg, Germany (H.S., R.E., D.H.)

The burden of noninteroperability between on-line genomic resources is increasingly the rate-limiting step in large-scale genomic analysis. BioMOBY is a biological Web Service interoperability initiative that began as a retreat of representatives from the model organism database community in September, 2001. Its long-term goal is to provide a simple, extensible platform through which the myriad of on-line biological databases and analytical tools can offer their information and analytical services in a fully automated and interoperable way. Of the two branches of the larger BioMOBY project, the Web Services branch (MOBY-S) has now been deployed over several dozen data sources worldwide, revealing some significant observations about the nature of the integrative biology problem; in particular, that Web Service interoperability in the domain of bioinformatics is, unexpectedly, largely a syntactic rather than a semantic problem. That is to say, interoperability between bioinformatics Web Services can be largely achieved simply by specifying the data structures being passed between the services (syntax) even without rich specification of what those data structures mean (semantics). Thus, one barrier of the integrative problem has been overcome with a surprisingly simple solution. Here, we present a nontechnical overview of the critical components that give rise to the interoperable behaviors seen in MOBY-S and discuss an exemplar case, the PlaNet consortium, where MOBY-S has been deployed to integrate the on-line plant genome databases and analytical services provided by a European consortium of databases and data service providers.

The evolution of data representation and analytical service provision in biology has been largely autonomous and ad hoc, resulting in the proliferation of an absurd number of standards for data formats and thousands of independently derived data analysis interfaces. Stein describes the current state of bioinformatics as "city-states…rival groups, each promoting its own Web sites, services, and data formats" (Stein, 2002). For example, there are at least 20 different formats for representing DNA sequences (http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Themes/SequenceFormats.html). The complexity of navigating these data retrieval and analysis networks is a bottleneck for bioinformaticians, and a near-insurmountable barrier to bench scientists who often have no programming skills. Thus, integration of bioinformatics data and tools is vital, but difficult (Chicurel, 2002), and has only begun to be addressed in any comprehensive way in the past few years.

Most bioinformatics data and tools are available through the Web. Accessing the Web requires no knowledge of specific query languages, but rather researchers find their data of interest through query by navigation (Karp, 1995), as they move from site to site, interacting with different interfaces to extract each different type of data. Increasingly, however, these Web interfaces are becoming the rate-limiting step for biological and/or bioinformatics analyses. Postgenomics experiments require access to dozens of data types for tens of thousands of data points simultaneously. This cannot be achieved with common Web-based tools; such analyses require programmatic access to Web interfaces such that large quantities of data can be pipelined from one interface to the next. Unfortunately, where they exist at all, most bioinformatic pipelines employ fragile screen-scraping methodologies to locate and extract data out of human-readable Web pages. Such pipelines are difficult to create, task specific, high maintenance, and error prone. Thus, the limitations of existing Web-based genomics and bioinformatics resources can be summarized as follows (discussed in more detail in Gribskov, 2003; Stein, 2003; Hernandez and Kambhampati, 2004; Schoof et al., 2004). The distributed nature of on-line data necessitates the manual collection and warehousing of this data to execute complex

[1] This work was supported by Genome Canada/Genome Prairie, A Bioinformatics Platform for Genome Canada (Dr. Christoph Sensen, Lead Principal Investigator). PlaNet is funded by EU Framework V (grant no. QLRI–CT–2001–00006).

* Corresponding author; e-mail mwilkinson@mrl.ubc.ca; fax 1–604 –06 –274.

www.plantphysiol.org/cgi/doi/10.1104/pp.104.059170.

queries and analyses. Heterogeneous data formats make data download and warehousing a tremendous effort. Knowledge about what data exists at each resource must be gained by personal experience. Heterogeneous Web output is difficult for humans to interpret and compare rapidly, and even more difficult for machines to accurately parse to extract data of interest. Web pages often contain information only about single data-points, and Web interfaces generally cannot operate on bulk data. The degree of integration on the Web is limited by the knowledge and time/resource investment of individual data providers, rather than the inherent properties of the data itself, and this is subject to budgetary constraints. The lack of interoperability between Web-based resources requires the discovery of related information through manual copy-edit-paste into the search interfaces of each Web resource. The lack of common vocabularies or term definitions for most data types prevent data arising from disparate sites from being immediately comparable.

Web Services provide a more robust, programmatic interface for Web-based tools that avoids screen-scraping (Stein, 2002) and are increasingly being used in biology and bioinformatics to automate execution of services. The Web Service paradigm involves defining the inputs and outputs of a service in terms of Extensible Mark-up Language (XML) primitives such as ''string'' and ''integer'' and then registering the existence of the service in a centralized searchable registry. While this aids in making the interfaces machine readable, it does not deal with the significant problem of data typing; paradigmatic Web Service interfaces could neither distinguish between a sequence in FASTA format or in EMBL format (they are both strings), nor even distinguish between a DNA sequence and a journal abstract. As such, they have failed to make a notable impact on the interoperability problem in the realm of bioinformatics.

THE SOLUTIONS

Four widely recognized interoperability architectures have been attempted: BioMOBY including both MOBY services (MOBY-S; Wilkinson and Links, 2002; Wilkinson et al., 2003, 2004; Lord et al., 2004) and Semantic MOBY (S-MOBY; Wilkinson et al., 2003; Lord et al., 2004) branches, ^{my}Grid (Goble et al., 2003; Stevens et al., 2003), and caBIO (http://ncicb.nci.nih.gov/core/caBIO). All four are based on Web or Web Services technologies and use an additional specification to describe the semantics of their data operations. MOBY-S uses a set of simple, end-user-extensible ontologies as its framework to describe data semantics, data structure, and classes of analytical service. These ontologies are shared through a novel Web Service registry system, MOBY Central, which uses the ontologies to semantically bind incoming service requests to service providers capable of executing them. S-MOBY stylistically resembles the Representational State Transfer architecture (Fielding, 2000) and utilizes Semantic Web technology to describe data semantics and structure. Services are registered using a novel semantic registry. ^{my}Grid supports many types of Web Services, including MOBY-S, but does not formalize any particular data semantics nor structure. ^{my}Grid ontologies (Stevens et al., 2003) describe the tasks that a service provider may perform on incoming data and/or the resources the service provider uses to perform these tasks. These are supported by a registry, underpinned by UDDI (http://uddi.org), but with a similar functionality to that of MOBY Central. caBIO relies on ontologies from the caCORE project from the National Cancer Institute of America. It defines both data structures and semantics through the provision of a programmatic interface (API) that is enacted through Web Services, and these are discoverable through a UDDI-based registry.

caBIO differs significantly from the other three projects in that its API is object-oriented; the data, and programmatic methods that can be called on that data, are encapsulated into the same ''object.'' Conversely, MOBY-S, S-MOBY, and ^{my}Grid focus primarily on passing lightweight data-only messages from service to service, without defining the operations that can be invoked on any given data object (Wilkinson and Links, 2002; Stevens et al., 2004). Though the caBIO approach gives more programmatic power, focusing on a data-only messaging system seems to be more flexible and requires less centralized maintenance. The two approaches, however, are largely complementary.

S-MOBY is currently at an early prototype stage of development; however, it is already showing great promise as an exceptionally rich yet flexible paradigm for Web Service discovery and invocation. However, since it has not yet been widely deployed nor tested, it will not be discussed further here.

MOBY-S and ^{my}Grid have both adopted a more traditional Web Service paradigm. This architecture relies on a registry (yellow pages) to store interface definitions, and a brokering API to mediate the discovery of registered services. In MOBY-S, this brokering function is carried out by the MOBY Central interface, while in ^{my}Grid the ''Feta'' interface fulfills this role (Lord et al., 2004). Both ^{my}Grid and MOBY-S employ ontologies in their brokering systems to assist in the discovery process by formalizing the way service inputs, outputs, and operations are described. This enhances the power and accuracy of searches done on the underlying registry. MOBY Central and Feta are sufficiently similar in their functionalities that an initiative is now under way to merge these into a single unified Web Service discovery engine for bioinformatics. This merger will dramatically increase the power available to bench scientists to pursue large-scale data mining and exploration with relatively little specialized training or computational infrastructure.

Gribskov stated four challenges for biological databases: integration, interoperation and federation; ontologies and defined semantics; community

annotation; and integration of analysis tools (Gribskov, 2003). Achieving all of these goals is difficult for any single database or project in isolation; however, several European plant genomics database providers recently initiated a collaboration to form the PlaNet project (http://www.eu-plant-genome.net) to address these issues (Schoof et al., 2004). PlaNet has chosen the MOBY-S architecture as the basis for its interoperability layer, and the implementation and consequences of this decision are described here.

THEORY: THE MOBY-S INTEROPERABILITY ARCHITECTURE

MOBY-S Architecture Overview

BioMOBY is an open source ontology-based bioinformatics interoperability research project established in late 2001. The MOBY-S branch of this project is currently being implemented by more than 70 service providers worldwide with a membership exceeding 140 individuals, most of whom do not explicitly coordinate their data sharing activities. (There are a variety of ways to explore the contents of the MOBY Central registry. Several of these are available at http://biomoby.org/toolstoys.html.) Rather, these data and analysis hosts simply adopt the simple, extensible standards for data representation required by the MOBY-S platform, and interoperability is achieved therein.

The target audience for MOBY-S is the amateur bioinformatician supporting a small- to mid-scale biological database or analytical service. Although MOBY-S is likely to be useful to large-scale projects also, the aim of making the technology simple enough for individuals or groups with limited bioinformatics experience and resources was a key requirement throughout the development process. As such, implementing the MOBY-S system as a service provider requires only limited programming skills and is supported by Perl, Java, or Python codebases available from the open-source BioMOBY code repository. MOBY-S data can be accessed by biologists entirely by mouse clicks with no need for additional programming.

The MOBY-S registration, discovery, and execution process is described in Figure 1. The registry stores service interface descriptions as provided by independent service providers (Fig. 1A); the MOBY Central brokering API accepts queries from data consumers (Fig. 1B) and in return provides them with the interface definition of appropriate services in the registry. The interface definition documents are fully machine readable, allowing the discovered service(s) to be invoked automatically (Fig. 1C) with singular or bulk input data.

MOBY-S differs from all other Web Service systems in one significant way: MOBY-S defines all valid data types in an ontology. This has both positive and negative consequences. It simplifies the problem of interoperability by limiting the possible range of interfaces that can exist; however, this comes at the expense of flexibility, since service providers cannot create arbitrary interfaces at will (as they commonly do with Web pages). Nevertheless, the system has been successfully deployed by one large international consortium, PlaNet, and in light of that success is currently being deployed by several additional national

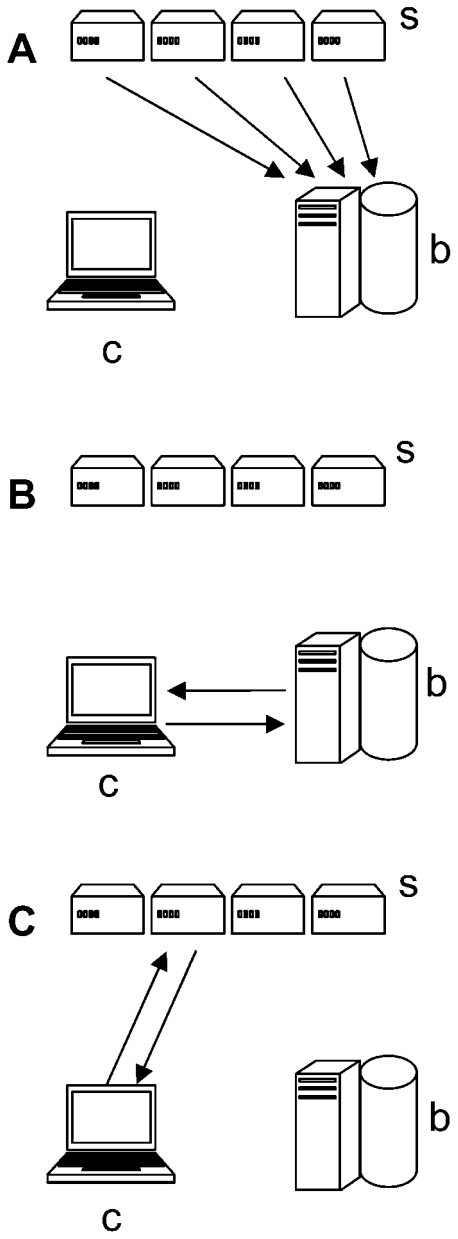

Figure 1. Schematic representation of the participants, the architecture, and the messaging processes in the MOBY-S Web Service brokering system. A, First, service providers (s) each register themselves in a centralized registry (b) indicating their input, output, and the type of data service they provide. B, Data consumers (c) then query the registry looking for service providers capable of executing the desired data retrieval or transformation service, the registry responds by providing a machine-readable description of the service interface. C, The data consumer is now able to automate the execution of the desired service to acquire the output data.

and multinational consortia, suggesting that these limitations are less significant to the data hosts than the power gained through interoperability.

Interoperability is achieved by making the MOBY Central service broker aware of the data-type ontology, enabling it to map incoming data onto services capable of consuming that data by ontologically reasoning over both the data-type and service definitions. Thus, data ''surfs'' from one service to the next, requiring neither explicit coordination between service providers nor any human intervention to reformat the service request. This Semantic Web on-the-fly behavior is unique to MOBY-S. Moreover, MOBY-S is capable of discovering and executing, unattended, a pathway through multiple independent Web Services that will derive a desired output data type from a starting input data type (http://www.ebi.ac.uk/collab/mygrid/service2/jmoby/graphs). This astonishing power has never been possible with any previous Web Service or semantically based bioinformatics system and heralds a new generation of bioinformatics data retrieval and analysis applications aimed directly at the biologist, capable of enabling bench scientists to execute complex and previously programmatically elaborate analytical workflows with just a few mouse clicks.

During the month of July, 2004 the public MOBY Central server responded to more than 100,000 service queries from 1,067 distinct sites around the world. A variety of third-party on-line tools have been built with embedded MOBY-S technology, including BioTrawler (http://llama.med.harvard.edu/cgi/BioTrawler), Gbrowse (http://www.gmod.org/ggb/index.shtml), and DragonDB (http://www.antirrhinum.net). In addition, support for MOBY-S data discovery has been incorporated into widely used standalone data retrieval and analysis systems, including Taverna (http://taverna.sourceforge.net/main.html) and BlueJay (http://bluejay.ucalgary.ca).

MOBY-S Data Syntax and Semantics

At the core of MOBY-S data structures is the MOBY Triple. The three components of the triple are: Namespace, the domain in which a data entity resides; Identifier, the specific data entity within that domain; and Class, the way a data entity will be represented. To explain how these components are used, we might consider an example using the PISTILLATA (PI) locus of Arabidopsis (*Arabidopsis thaliana*) as described by the Arabidopsis Genome Initiative (AGI).

The AGI locus Identifier for *PI* is At5g20240. The Namespace within which this Identifier is to be interpreted is designated AGI_LocusCode in the MOBY-S system (namespaces in BioMOBY are borrowed from the Gene Ontology Cross Reference Abbreviations List, and a complete list is available for download or browsing http://mobycentral.cbr.nrc.ca/cgi-bin/XrefAbbs). Thus, the Namespace and Identifier, together, are sufficient to uniquely refer to any data entity in any database, even if the identifier itself is nonunique over multiple databases. The Namespace also provides some indication of what the identifier means (its semantics). In this case, the AGI_LocusCode Namespace indicates that the Identifier should be interpreted as a locus from the AGI, and the Identifier At5g20240 describes which locus, *PI*, is being described. The third component of the triple, the Class, is an indication of which properties of that data entity are going to be described in the data object. If the nucleotide sequence of PI were of interest, the Class DNASequence would be appropriate. The resulting triple is shown in Figure 2A. Identifiers should follow the formatting rules, including case sensitivity, indicated by the assigning Namespace authority, and similarly service providers may reject identifiers that are improperly formatted.

The Class portion of the triple has an important additional role. Valid classes are defined in the MOBY object ontology (Fig. 3). The object ontology defines the constitution and, thereby, the syntax or data format for representation of each data-type. In this example, DNASequence is a node in the object ontology in which the data class has an integer component representing the length (inherited from VirtualSequence) and a string component representing the sequence. Represented as XML, the DNASequence object corresponding to *PI* is shown in Figure 2B. The same entity could also be represented as a FASTA object, which corresponds to a node in the object ontology that inherits from text formatted. The XML schema of this node can be derived from the object ontology and the XML representation would be as shown in Figure 2C.

An issue that often arises in biology is that of nonunique identifiers. Although from an informatics perspective an identifier must, by definition, be unique, many of the entities commonly used as identifiers in biology are nonunique. Gene names are a perfect example of this. For example, in Arabidopsis, there are three unrelated genes named ADK1. These have been assigned the AGI locus codes At1g03930, At1g09820, and At5g63400, and, therefore, each of these

A DNASequence , AGI_LocusCode , At5g20240

B
```
<DNASequence namespace="AGI_LocusCode" id="At5g20240">
        <Integer namespace="" id="" articleName='length'> 916 </Integer>
        <String namespace="" id="" articleName='SequenceString'>
        aaaattgggaaagggaacgatagagaaagatgggtcgaggaac
        gaccgcgataacgaggatagagaaagcaaaaaacagagtggt…
        </String>
</DNASequence>
```

C
```
<FASTA namespace="AGI_LocusCode" id="At5g20240">
        >At5g20240
        aaaattggaaagggaacgatagagaaagatgggtcgaggaacgaccgc
        gataacgaggatag agaaagcaaaaaacagagtggt…
</FASTA>
```

Figure 2. A, The MOBY Triple representing a DNASequence object containing the AGI Locus At5g20240. B, The XML serialization of the same object, derived from an interpretation of the object ontology (Fig. 3). C, The same data entity now represented as a FASTA object, as defined by the object ontology (Fig. 3).

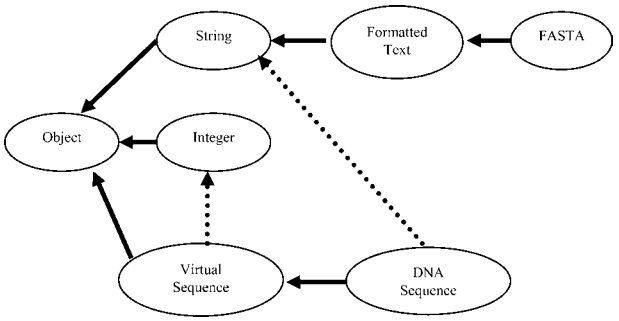

Figure 3. A small portion of the MOBY-S object ontology. The ontology has object classes at the nodes, and these are connected by two types of edges representing subclass (''ISA''; solid arrows) and container relationships (dashed arrows). The graph above would be interpreted using the following statements: ''A string is a type of object. An integer is a type of object. A virtual sequence is a type of object that also contains an integer. A DNA sequence is a type of virtual sequence that also contains a string. Formatted text is a type of string. FASTA is a type of formatted text.'' The full object ontology can be obtained as an RDF-XML document from http://biomoby.org/RESOURCES/MOBY-S/Objects.

genes has a true unique identifier that would be the preferred means of referencing the locus in a MOBY context. Nevertheless, it would be unreasonable if a system designed to aid biologists was incapable of communicating about biological entities using their natural terminologies. As such, there is no strict limitation on MOBY namespaces and identifiers to be unique identifiers, and certain namespaces such as Global_Keyword and DragonDB_LocusName are assumed to have the potential for identifying multiple entities. Service providers who consume these types of identifiers as input are free to return multiple responses as appropriate. It should be noted, however, that a proliferation of such nonunique identifiers is highly destructive to the ability to automate analytical pipelines, and thus, true unique identifiers should be created by data hosts and used by biologists whenever possible.

Knowledge of the valid namespaces, both in their format and their meaning, might pose a barrier to the initial use of MOBY-S by biologists, particularly as the number of namespaces grows (currently MOBY-S has 162 registered namespaces; http://mobycentral.cbr.nrc.ca/cgi-bin/types/Namespaces). This problem is being addressed by the ongoing creation of a variety of search and exploration tools for the namespace ontology, but since the concept of a namespace itself is quite foreign to our target audience this barrier remains a significant one for new users of the MOBY-S system. It is likely that most namespaces within biologists' immediate domain of interest are already tacitly or explicitly known to them (for example, most Arabidopsis researchers are likely already aware of the existence of AGI locus codes) and since it is only the initial approach to the MOBY-S system that requires human interaction, subsequent service invocations being orchestrated by machines, there will seldom be a requirement for biologists to have intimate knowledge of a broad range of namespaces beyond those that are already familiar to them. As such, it seems likely that the combination of user-friendly interfaces and exploration tools currently under construction will suffice to assist new users in their early usage of the MOBY-S system.

Two notable observations should be made from these examples. The first is that the syntax and the semantics are kept separate in MOBY data; that is, the essence of a data entity is independent of the way it is represented. This allows the MOBY system to represent the same data in a variety of ways, and this is a critical behavior given the necessity of moving data from one service provider to another where the different services consume/produce different data formats. The second is that the precise syntax of a data type is defined by an ontology; in fact, the MOBY object ontology itself is a novel XML schema definition. The XML representation of any class in the ontology can be determined solely from its position within the ontology. Thus, it is possible for end-users to define their own data types simply by registering new ontology nodes, without having to understand the XML Schema Definition language (XSD). This was a critical requirement for a system that is intended to be used by amateur bioinformaticians, since XSD documents are nontrivial to construct.

The object ontology is key to MOBY-S' interoperability behavior, and its simplicity has resulted in a proliferation of new object definitions by MOBY-S service providers. The object ontology allows service providers to define their Web Service inputs and outputs in simple terms, often simply by naming the appropriate ontological nodes. This enables the discovery of services based on their inputs/outputs as defined in these simple terms; yet the terms resolve to machine readable specifications of the precise structure of the service interface such that the process of invoking the service can be fully automated.

Finally, it is interesting to note that the object ontology arose from a community-driven effort. Only the first dozen data types were created in a centralized way by the core BioMOBY development team. It was then released to the community with an open, public API for registration of new data types, as required by service providers. The object ontology now consists of 113 data types (as of February, 2005; http://mobycentral.cbr.nrc.ca/cgi-bin/types/Objects), spanning sequence-related objects such as FASTA, nucleotide and amino acid sequences, images, SNP and haplotype data, germplasm data, legacy flat-file formats such as Blast and GenBank, and even novel extensions of legacy file formats such as annotated images. As new service providers plug in to the MOBY-S network, they add new data types to the ontology that eventually benefit all users by enhancing the scope of data available via the MOBY-S system, as well as reducing the barrier of entry to the system by expanding its native scope. This community-led ontology development path, originally arising out of financial necessity, was by far the most risky aspect of the BioMOBY project since, unlike the

World Wide Web itself, the object ontology mandates a certain degree of logic in its structure and would not tolerate widespread nonsensical registrations (though a certain number of these do exist and are removed by manual curation from time to time).

The MOBY approach to ontology creation is even more open than the notably successful Gene Ontology (GO) project, where new ontology terms are first passed through a curatorial process before being allowed into the production ontology. Nevertheless, the success of the MOBY-S object ontology shows that resources, even at this level of complexity, can be created through distributed, noncoordinated, and minimally curated community efforts. It is likely that this success springs from the same root cause as was proposed for GO; that is, the existence of an intelligent and self-interested user base (Lewis, 2004). The need for an interoperability system like MOBY-S was (and is) dire, and it is in the interests of all involved parties to use MOBY-S in the way it was intended. As such, there is no incentive to register useless data types, and strong incentive (and reward) for maximizing interoperability by registering well-conceived, widely applicable data types.

MOBY-S Service Types

In MOBY-S, service providers consume ontologically defined data types, manipulate them in some way, and then return ontologically defined data types. The valid types of manipulation that can be executed by MOBY services are defined in the service ontology. The service ontology is a simple hierarchy with several roots including parsing, analysis, retrieval, and registration. It defines all possible operations that might be executed on incoming data and, like the object ontology, may be extended by new service providers to include new types of analyses.

The most appropriate service ontology term is selected by the service provider when they register their service in MOBY Central. The service ontology can be similarly used during service queries, where the client either asks for specific types of service operation or may select more shallow ontology nodes (i.e. more abstract types of service operations) in order to discover a broader range of service providers. For example, rather than specifying ''Blast'', ''Fasta'', or ''Smith-Waterman'', the service ontology term ''Alignment'' might be chosen in a service query to discover all service providers that execute any type of sequence alignment algorithm.

MOBY Central

The final component of the MOBY-S system is the registry itself. MOBY Central does not store any biological data, but is capable of exploring the ontology of biological objects and service types to more richly respond to requests for service discovery. MOBY Central's interface can respond to a variety of common queries such as request for service discovery by input data type, output data type, or the type of analysis the biologist wishes to execute (i.e. ''What can I do with this data?''; ''Who can give me this data?''; or ''Who can provide this analytical service to me?'').

Service providers register a simple description of their service interface, input data type, output data type, and service type, in MOBY Central, and are returned a formal service signature document that will be used by MOBY Central to poll the service in the future. Similar to traditional Web search engines, MOBY Central regularly retrieves these service signatures back from individual service providers and compares them to the information in the registry. Updating service registration therefore can be accomplished by editing the service signature file on the service provider's local server and then either prompting MOBY Central to repoll the service immediately or passively allowing MOBY Central to discover the changes on its next update cycle. Similarly, removing a service from the registry can be accomplished by deleting the service signature file. While this does not entirely solve the problem of dead services, a problem that has not been solved by any Web technology, it does at least necessitate that the service providers Web server is active to respond to the MOBY Central poll, and thus truly dead services will be automatically removed from the registry after a short time.

What MOBY-S Is Not

Though its functionality often makes MOBY-S appear to be a distributed query system, it is not. In particular, it lacks the ability to (natively) support Boolean queries such as NOT, OR, and AND, and this is often pointed out as a weakness of the MOBY-S platform versus warehoused or tightly coupled federated databases, where arbitrary, rich queries can be executed using a query language such as SQL. MOBY-S addresses a different problem; it attempts to provide a mechanism for data and analysis integration, a task that is regularly undertaken (in a limited way) by the providers of a data warehouse or federated database. However, instead of having to transform all data to the common warehouse schema before any query can be executed, MOBY-S provides a framework where integration can happen on-the-fly, allowing a user to tap into the most recent data from a number of different databases and combining it in novel ways that could not have been foreseen by a warehoused system. MOBY-S interoperability is limited only by the altruism of its many service providers and therefore can federate a much wider range of data with less centralized effort. Unfortunately, MOBY-S currently lacks a versatile query tool that allows rich queries to be executed on the federated data, such as ''Retrieve the phenotypic images of all mutations in genes with GO annotations of 'apoptosis' and 'cell membrane'''; however, such tools are actively being developed. Nevertheless, certain Boolean operations can be achieved in

MOBY-S in a manner similar to that in AceDB (Durbin and Thierry Mieg, 1991), where independently derived keysets are available for set-operations such as union and intersection. Union and intersection are two terms found in the MOBY-S service ontology, and services have been created specifically for the purpose of executing AND and OR Boolean functions on sets of outputs from disparate service providers. Similar operations, and many more complex processors and filters, are available within Taverna (see below). Thus, MOBY-S, while not implementing a rich distributed query system at this time, is not as limited as proponents of federated databases might suggest.

There is neither explicit requirement for, nor support for, wildcard matching by MOBY-S service providers. For example, a service provider is not required to support incoming requests for data of the form ''AGI_LocusCode:At5g202*''. There has been limited discussion among the BioMOBY development community as to whether wildcard support should be described as part of the service signature, but to date no action has been taken. This is probably due to the fact that wildcard support is primarily designed for human-readable interfaces, while MOBY-S is primarily designed to be machine-readable and executable; i.e. from the perspective of the machine, it is as straightforward, more consistent, and more accurate, to pass lists of identifiers rather than a single identifier with a wildcard. Nevertheless, it is certainly not invalid to construct query invocations using wildcards, and individual service providers may or may not chose to support these types of queries. At this time, however, this information is not explicitly captured as part of the service registration and therefore could not automatically be detected and invoked as part of an automated analytical pipeline.

Similarly, MOBY-S is not a quality control system. The accuracy and quality of data is, as with all Web interfaces, the responsibility of the data host. This is, however, a particularly strong concern in the context of an automatable system such as MOBY-S where a biologist becomes several steps removed from the process of data retrieval and analysis. While MOBY-S cannot guarantee data accuracy, it does provide a means for limiting the choice of discovered service providers either by their domain name (e.g. mips.gsf.de) or by the service provider's claim to be ''canonical'' for the data or service offered. As such, it remains within the power of the biologist to select data and service providers that they prefer or trust, and they are encouraged to do so.

PRACTICE: FEDERATING THE PLANET DATABASES USING MOBY-S

The PlaNet consortium includes Arabidopsis stock, genome, and research databases in Germany, Belgium, England, France, Spain, and The Netherlands. All PlaNet partners are currently being made interoperable through a MOBY-S infrastructure. When new data or analysis services arise at any of the individual participating centers, they simply register the existence of this service with the MOBY Central installation hosted at the Munich Information Centre for Protein Sequences (MIPS; http://mips.gsf.de) and it immediately becomes available to all other partner centers and software tools, with no further coordination or reprogramming required.

Thus, the bioinformatics nation proposed by Stein is starting to be realized through the MOBY-S project. We will discuss here what we have discovered about the nature of the bioinformatics interoperability problem in practice, the aspects of the MOBY-S solution that make it unique and successful for our task, and what challenges remain to achieve the final goal of a fully integrated domain of biological knowledge.

PlaNet: Aims and Architecture

PlaNet is a distributed effort among bioinformatics groups and plant molecular biologists to establish a comprehensive integrated database in a collaborative network. This approach diminishes the strain on limited individual resources through distribution of the curatorial burden, while at the same time maximizing the value of the independent data collections. It also creates a nucleus for other European and international groups and consortia to join and utilize the network.

Overall objectives of PlaNet are to: capture genomic information into a comprehensive platform; establish a network of dynamically interconnected European plant databases; develop new methods for data exchange, database integration and access; provide high quality integrated data resources for research; ensure high availability of data generated by European laboratories and plant research consortia (data platform); incorporate expert knowledge and regional networks; focus direct contribution by regional plant research communities (expert annotation system); perform systematic classification of plant genes and regulators; and develop standards for data representation and nomenclature.

PlaNet aims to provide a comprehensive plant genomics data platform allowing access to integrated data from distributed sources. One approach toward data integration is warehousing, but this has severe drawbacks (Stein, 2003). All data must be unified in a common database schema, and all data regularly transformed and imported. However, as our knowledge increases, database schemas evolve to accommodate new data types or biological relationships, and it is extremely hard to incorporate all these changes into the warehouse. A federated database allows the individual databases to continually work on their data representation, as long as they keep standardized interfaces intact that at least allow part of their data to be integrated.

The overall structure of the PlaNet federated database is shown in Figure 4. The data layer consists of the specialized partner databases: MAtDB (http://mips.

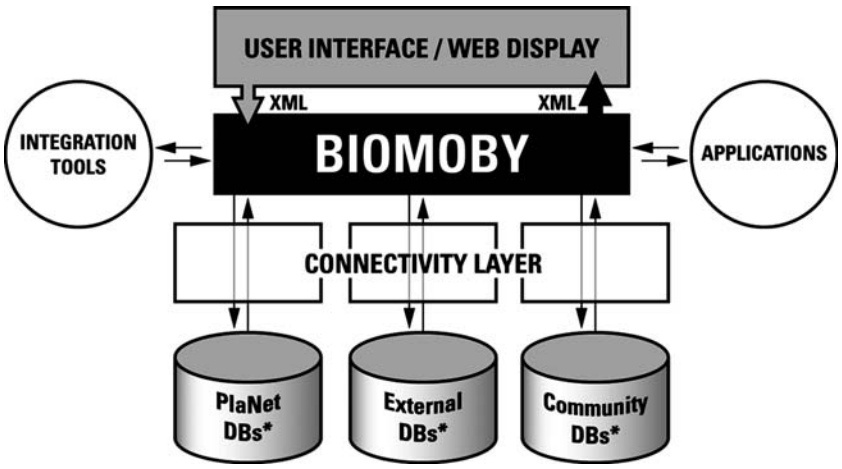

Figure 4. Architecture of the PlaNet federated database. The user interface and Web display provide a single point of access to data and analysis tools provided by the distributed partners. To this end, all databases are linked via a connectivity layer developed by PlaNet in collaboration with the BioMOBY project. The component databases make their data available as BioMOBY Web Services (the connectivity layer) and interactions between individual services, and between services and the user interface, are orchestrated by the BioMOBY registry system (BioMOBY layer). Integration tools can be developed that use the connectivity layer to compare data in different databases and thus ensure data consistency. Besides data sources, BioMOBY also provides for the integration of analysis tools and applications.

gsf.de/proj/thal/db/), NASC stock catalogue (http://arabidopsis.info/), NASC arrays (http://affymetrix.arabidopsis.info/narrays/experimentbrowse.pl), AtIDB (http://atidb.org/cgi-perl/index), PlantCare (http://intra.psb.ugent.be:8080/PlantCARE/), WAtDB (http://www.watdb.nl/), rRNA-DB (http://www.psb.ugent.be/rRNA/), FlagDB (http://flagdb-genoplante-info.infobiogen.fr/), and public databases like the EMBL sequence database integrated through one of the partners. A common connectivity layer standardizes access to data and allows clients (e.g. Web interfaces or analysis tools) to access all data through this common layer. BioMOBY was selected for the implementation of the connectivity layer. Though all PlaNet MOBY-S Web Services are accessible through the public Gbrowse_moby Web-based user interface (Wilkinson, 2003) distributed with the Gbrowse software package (Stein et al., 2002), PlaNet-specific clients, like the AGI locus report, query multiple databases in parallel through MOBY-S Web Services and allow access to the data of all PlaNet partners through a single Web page (see below).

Implementation of MOBY-S

The connectivity layer is realized with BioMOBY's MOBY-S Web Service specification. The PlaNet project uses international standards where feasible, e.g. the GO (Ashburner et al., 2000) or Sequence Ontology (http://song.sourceforge.net), and avoids creating project-specific vocabularies or standards with an eye to future expansion and integration of the project beyond PlaNet.

Data objects are modeled on internationally accepted schemas where available, e.g. MAGE-ML (http://cgi.omg.org/cgi-bin/doc?lifesci/01-10-01) for transcriptomics data, Generic Feature Format (http://www.sanger.ac.uk/Software/formats/GFF/GFF_Spec.shtml), or GAME-XML (Lewis et al., 2002) for genome annotations or NCBI BLASTXML (ftp://ftp.ncbi.nlm.nih.gov/blast/documents/xml) for BLAST results. This is possible because the MOBY-S object ontology allows inheritance from a class "text-xml" that enables the addition of legacy XML data models into the MOBY-S system. When developing new MOBY objects, care was taken to extend existing object models wherever possible. In this way, interoperability is maximized, since more complex objects can be consumed by services with less complex input requirements, so long as there is an ISA relationship between the incoming (complex) MOBY object, and the required (simple) MOBY object. The current data objects defined and used in PlaNet can be retrieved from the PlaNet homepage (http://www.eu-plant-genome.net/) under the link to tools.

MOBY-S services were implemented as required by specific use cases. As a first step, PlaNet chose the task of enabling all data on an AGI locus code to be retrieved from all distributed databases and displayed in a single Web page. AGI locus codes (Schoof et al., 2002) are unique identifiers for a genomic locus in the Arabidopsis genome that are maintained by The Arabidopsis Information Resource (TAIR; Rhee et al., 2003). These are used by many data resources and publications and are registered as a namespace in MOBY Central. Using these codes, data on genes and proteins encoded by that locus can be retrieved, e.g. gene models, expression, protein sequence, protein function, or phenotypes of knockouts. Figure 5 schematically shows a subset of the information retrieval pathways surrounding AGI locus codes that have been implemented in the PlaNet MOBY-S interoperability layer. To access all this information from a single Web page, i.e. without using the search interfaces of several databases, the AGI locus report was implemented.

Example: AGI Locus Report

To show the variety of data that can be retrieved through PlaNet with a single input, the AGI locus report was implemented (see Fig. 6). This application retrieves all basic information available within all PlaNet databases for a given AGI locus code. On the

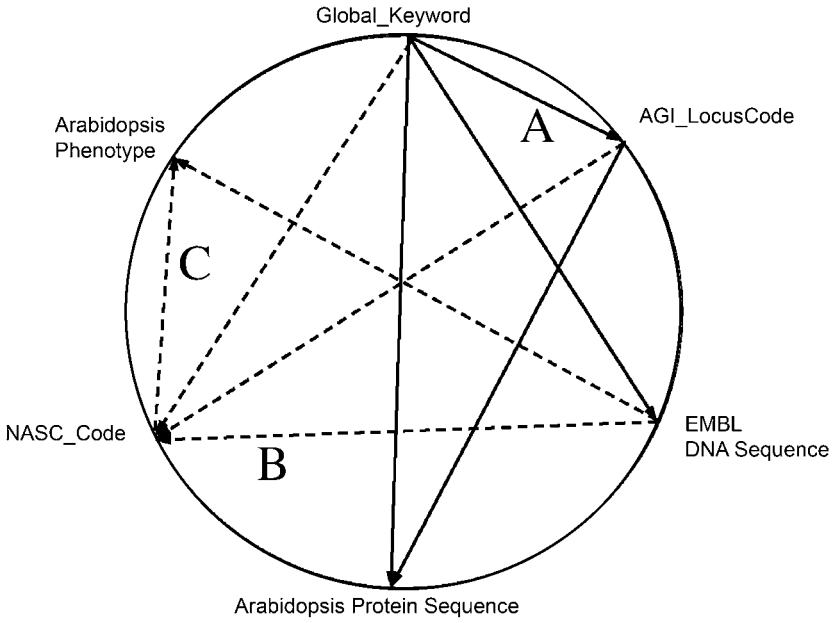

Figure 5. Examples of data flow made possible by the PlaNet BioMOBY connectivity layer. Services (not all shown here) are represented as arrows linking input and output data objects in this graph. For example: A, starting from a keyword all corresponding AGI locus codes can be found. B, The AGI locus codes can then be used to retrieve the NASC codes (http://www.arabidopsis.info). C, With the NASC code in hand, a third service can be queried to find the Arabidopsis phenotypes for them. This example combines three services that operate on data from two different service providers (MIPS/Neuherberg, represented as solid arrows, and NASC/Nottingham, represented as dashed arrows), previously requiring two separate Web query forms. The workflow just described could also be automated in an application. Complex queries can be realized through pipelining services and filters in a workflow where the output of one service is the input of another service. The services are accessible through a simple, Web-based query client prototype at http://www.eu-plant-genome.net.

one hand, it provides links to the relevant report pages of several database Web interfaces that display comprehensive data from that site. This represents a basic extension of the "query by navigation" method of exploring data. On the other hand, it utilizes the discovery capabilities of MOBY-S to display all Web Services that retrieve specific data for that AGI code directly from the distributed databases.

In order to not frustrate users with links or Web Services that lead to no data, the given AGI code is first

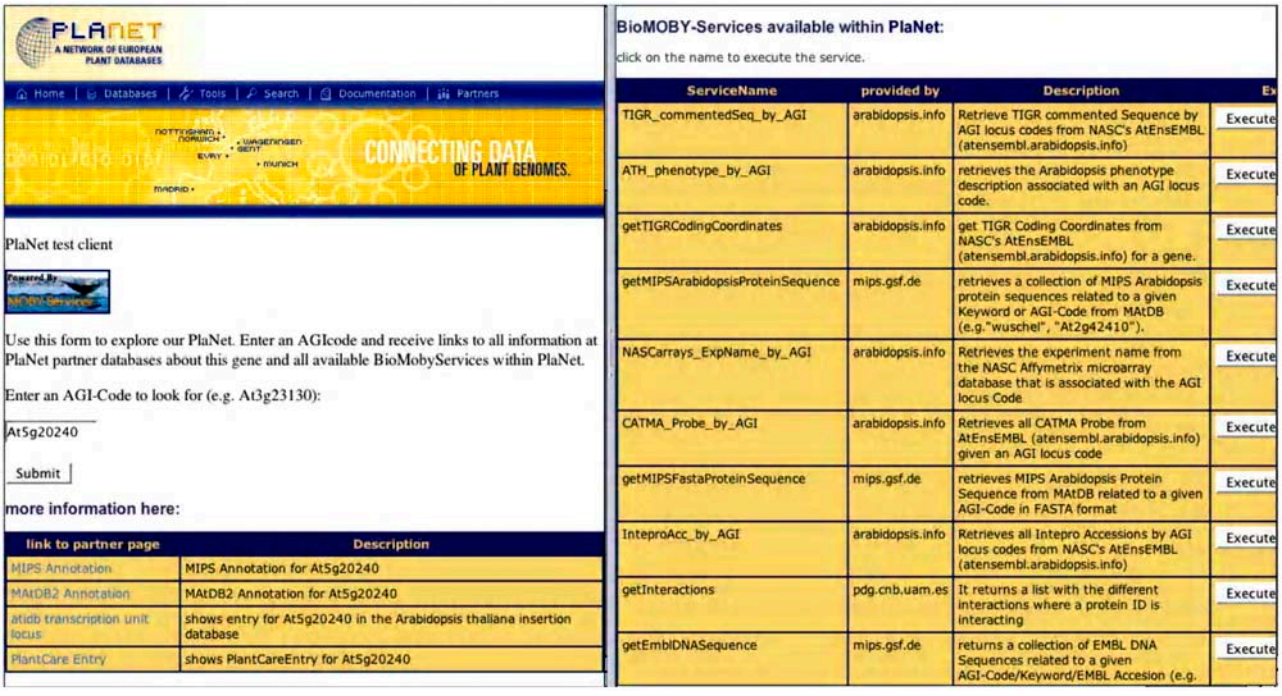

Figure 6. A sample AGI locus report Web page dynamically generated by querying the MOBY Central registry at MIPS. The AGI locus report gives links to the Web interfaces of partner databases (left) as well as all MOBY-S services (right) on the same results page. The list of services is not static, and will grow as PlaNet partners register new relevant services in MOBY Central. The end-user, however, simply chooses the data that they are interested in retrieving, and the transaction is automatically carried out for them, regardless of which partner(s) host the requested data. Thus, the AGI locus report page acts as a dynamic portal through which all information known about a locus can be retrieved.

checked for availability at the remote sites, and links are only displayed if data for this AGI code is available at a partner's database. Using a traditional Web interface, such a list could only be generated by maintaining a local list of AGI codes that are valid for each remote database. This clearly demonstrates an advantage of BioMOBY technology. Using MOBY-S services, a local list of valid locus codes is unnecessary; Discovered services can be executed on-the-fly and the AGI locus report client can easily omit services that return no data from its output page. This removes the curatorial burden of storing and maintaining local information about remote resources.

The list of executable MOBY-S services (Fig. 6, right) is generated dynamically from a query to MOBY Central that retrieves all services capable of consuming AGI locus codes as their input. These services are then displayed in a list and can directly be executed. The advantage of using the MOBY Central registry rather than traditional static interfaces is that this list is always up-to-date; as soon as a service is registered or deregistered by a third-party partner, this information will be reflected in the locus report Web page. In contrast, links to remote Web pages in traditional interfaces must be maintained manually.

Further development of the AGI locus report includes parallelized execution of selected services, which would directly display the output of MOBY-S services to the user instead of the currently displayed list of services. At the moment, this function has been deactivated, as serial execution of all services takes too long. An additional enhancement that has already been tested is to include pipelines of services; for example, as shown in Figure 5, there is currently no single service to retrieve phenotypes for an AGI locus code. However, by first retrieving the NASC code corresponding to an AGI locus code, the phenotype can then be retrieved by NASC code. The AGI locus report will include such pipelines that directly lead to important results that are not reachable in a single step.

To date, 97 services have been implemented, of which 84 are retrieval services that, for example, retrieve sequences, AGI locus codes, microarray experiments, gene annotation, interacting proteins, etc. Twelve services represent analysis services that perform BLAST or TargetP predictions (Emanuelsson et al., 2000) or parse/extract data. These services were deployed with minimal formal training in the MOBY-S system. Help was obtained as required from the BioMOBY mailing lists and by following on-line tutorials and code examples.

With this number of services, numerous possible connections between services from different database providers are possible, allowing users to browse data across the borders of individual data resources. While the Web interface is not yet as neat and intuitive as local hard-coded database interfaces, it would take considerably more effort to accomplish the same complexity and diversity of data by browsing the individual Web sites using copy-and-paste between the different local search pages. As such, MOBY-S has dramatically simplified the problem of integrating these disparate resources. However, at the present time, this is mainly appreciated by the data managers within the Project whose tasks have been greatly simplified; for an uninitiated Web user, the current user interfaces such as Gbrowse_moby and the AGI locus report are far too limited to showcase the full power of the interoperability provided by MOBY-S. More advanced Web interfaces are currently under development by both PlaNet and MOBY-S project members.

The completely new dimension of distributed analysis made possible by MOBY-S is better demonstrated by showcasing an application that can pipeline services from multiple remote sites into workflows executed at the click of a button. Taverna (Oinn et al., 2004) is a bioinformatics workflow-design and execution application that supports MOBY-S services. Taverna enables end-users to connect Web Services into a workflow/pipeline, which can be saved and rerun at any point in the future, using any input dataset. Taverna operates largely unattended. It is capable of detecting when a service has returned multiple versus single outputs and will iterate over each of these outputs, sending each one in turn to the next service and beyond. It is capable of pausing one branch of a workflow to allow another branch to complete in cases where input from two branches is required by the subsequent step. Processors are available to combine the output from both branches, e.g. using the union (removing redundancy), difference, or intersection. In addition, it detects and recovers from errors. As such, complex workflows can be loaded/designed/edited, and initialized by the researcher, and then left to run on their own whether for minutes, hours, or days. Output data and intermediate results are available for browsing and/or saving to disc when the workflow has completed, e.g. as Excel files for further analysis and visualization, and a full record of the provision of this data (e.g. database versions, software versions, dates, times, etc.) is provided for publication purposes.

To demonstrate a use case that would be tedious using preexisting Web interfaces, a workflow will be conceived through which all Arabidopsis proteins annotated with a given keyword shall be retrieved, together with all Arabidopsis sequences that show significant homology to these annotated sequences. In this workflow, once provided with a keyword, all proteins that are annotated with this keyword in any of the databases connected in PlaNet are retrieved; their sequences compared against all Arabidopsis proteins using BLAST with a cutoff; the AGI locus codes of matches better than the cutoff are extracted from the BLAST result; these are combined with the list retrieved by the keyword; the combined list is returned to the end-user. In practice, this workflow, using the keyword drought resistance, takes 50 s and retrieves 50 proteins at a cutoff selecting BLAST

matches with an e-value better than E-13. Formerly, this would require the researcher to: (1) execute a keyword search using the Web interface of a genome database, returning six AGI locus codes; (2) use links to navigate to the report page for each of the six loci and retrieve the protein sequence; (3) paste each of the protein sequences into BLAST input forms, possibly requiring reformatting of the sequence; (4) copy and paste the matches above threshold into some text editor; and (5) remove redundant codes, e.g. by using the sort function of the text editor. The advantages of the MOBY-S dynamic-discovery and automated-execution system are, therefore, clear both in reducing the time and effort researchers need to invest in data accumulation, as well as reducing the level of expertise and experience they require to discover the data at all.

Using a tool like Taverna, the resulting list of AGI locus codes could be analyzed further, e.g. by adding a service to the workflow that retrieves knockout mutants or their phenotypes for each of them. These extensions are almost effortless and again highlight the simplicity that is achieved by moving away from traditional Web interfaces toward a more interoperable system such as MOBY-S. Complex workflow examples are available from the PlaNet Web site (http://www.eu-plant-genome.net), e.g. implementing questions like ''What Arabidopsis genes from the NBS-LRR family, as defined by a Phytoprot (Louis et al., 2001; Mohseni-Zadeh et al., 2004) cluster, also are annotated with the keyword 'disease' and are associated with the plant ontology term 'leaf' ''?

DISCUSSION

An important design decision in the PlaNet architecture was the placement of the human-readable Web interface and the MOBY-S layers within the computational infrastructure. It is clear that the Web has become the primary means by which researchers gather their data, and as such, it is critical to maintain both human readable Web interfaces as well as the machine-readable Web Services. In principle, MOBY services could function by querying existing Web pages, converting the output into MOBY objects and passing these to the user; however, this architecture is inherently unstable. Web pages are specifically intended to evolve over time, and such evolution would necessitate the rewriting of all MOBY services each time the Web page was modified. The interoperability architecture we chose, however, is not prone to these problems since the interoperability layer underpins the visible data representation layer. All data retrieval and manipulation occurs in the MOBY service, and this is then passed back to the requesting machine, either to be passed onto another service (e.g. in a workflow) or to be examined and marked up for human readability on a Web page. Changing the markup, or increasing the number of services represented on the human-readable page, does not affect the service, and thus the primary data retrieval layer is highly stable. This architecture provides an enormous degree of flexibility compared to traditional CGI Web interfaces, or the increasingly common PHP (Pre-Hypertext Processor; http://www.php.net) interfaces that push these two distinct layers even closer together.

The consequence of this decision is that considerable effort was invested into local database infrastructure to facilitate the implementation of MOBY-S wrappers that connect data sources to the connectivity layer; however, this effort is both unavoidable and worthwhile. Multilayer architectures and separation of data model and presentation simplify and improve long-term maintainability of any data interface. The flexibility of MOBY-S was extremely important in this context, as interfaces providing at least some degree of interoperability could be implemented extremely rapidly by focusing on common ground first and leaving problematic cases for later. For example, interoperable access to phenotype data was implemented simply by creating a novel MOBY object ''PhenotypeDescription'' that contains nothing more than free text. Over time, these can be replaced by semantically richer, structured, and ontology-based objects as these elements become implemented and annotated in the source databases.

Similar to the PhenotypeDescription object above, support for legacy systems within the PlaNet consortium was achieved by creating MOBY objects that simply contain the XML documents, verbatim, as used by existing database interfaces. In this way, for example, all gene model annotation in MIPS plant genome databases (Schoof et al., 2002) can now be retrieved through MOBY-S services: The XML generated by preexisting middleware interfaces is simply enclosed (wrapped) in a text-xml data object and can be dynamically discovered, and passed, like any other piece of data in the MOBY-S system.

Though there is a publicly visible and open instance of the MOBY Central registry that serves the global bioinformatics community (http://mobycentral.cbr.nrc.ca), the MOBY Central discovery system was implemented de novo for the PlaNet consortium. This decision was made due to a number of limitations we observed in the brokering API. In particular, though simple data retrieval services are easily described, registered, and discovered, the service ontology of MOBY Central is not sufficiently rich to fully describe analytical services such as the BLAST or Primer3 algorithms, nor is it sufficiently rich to describe the underlying resources being used by a service provider. For example, a search for BLAST services on the global registry discovers a large number of BLAST servers, where the PlaNet BLAST service that uses only Arabidopsis genome data is one of many. By implementing MOBY Central specifically for the PlaNet consortium, the relevance of discovered services could be greatly enriched for consortium members. Through the ongoing collaboration with ^{my}Grid, who have designed an extremely rich ontology for describing Web Services,

this limitation of MOBY Central is being addressed, and the search and discovery mechanisms in future iterations of MOBY should be much richer and more focused (Lord et al., 2005).

As a next step, it is essential that the accomplished interoperability between the data resources is translated into more powerful query pages and more informative result pages for the Web user. One task in work is an AGI locus report that will directly display a summary of the most important data retrievable through PlaNet MOBY-S services and that could replace the gene report pages of the PlaNet Arabidopsis databases. This will be extended by an interface for configuration by the user, allowing selection of what data should be displayed in the report.

CONCLUSION

MOBY-S provides a straightforward, practical solution to the technological problem of interconnecting data provision and analysis resources. It is rapidly implemented (more than 80 interoperable services built for PlaNet within a period of several months) and provides an extremely flexible and resilient architecture upon which large data-sharing consortia can be built. Within the PlaNet consortium, it has proven to be sufficiently rich for most bioinformatics data provision problems; however, there are limitations to the richness of the discovery layer when they are applied to analytical services, and this required the consortium to implement their own copy of the MOBY Central registry rather than simply create and register services in the public registry. These limitations are actively being addressed by the BioMOBY developers and will be less severe in future iterations of the project. Nevertheless, MOBY-S has demonstrated sufficient power to enable previously nonexistent functionality for users and thereby facilitate research. As such, it has proven to be a valuable approach to facilitating resource sharing in large-scale collaborations and consortia.

Though the MOBY-S system itself has considerable power and flexibility, end-user experience is still hampered by limited, prototype client applications and the lack of sophisticated interactive interfaces in existing clients. More powerful client programs such as Taverna are now beginning to appear, and as such MOBY-S is becoming a viable option for widespread use, being proposed for several large projects such as the international genome sequencing projects for medicago and tomato and the Generation Challenge Program of the Consultative Group for International Agricultural Research (CGIAR). Thus, MOBY-S seems poised to become an important ingredient for next-generation databases and bioinformatics tools, offering new possibilities to both bioinformatics programmers/analysts as well as researchers browsing Web interfaces.

The future of MOBY-S within PlaNet will focus on significantly increasing the available services, increasing the scope of data types made available over the PlaNet MOBY-S network and identifying missing link services that are essential to enable traversal from one data type to another. Simultaneously, richer and more intuitive interfaces will be explored that can bring the full richness of the MOBY-S interoperability system to the bench scientist.

ACKNOWLEDGMENTS

M.D.W. thanks all members of the BioMOBY project for their enthusiasm and efforts, in particular Matthew Links, Bill Crosby, Martin Senger, Tom Oinn, Lincoln Stein, and his team at Cold Spring Harbor Laboratory, and Damian Gessler and his team at National Center for Genome Resources. Thanks also go to Carole Goble and Philip Lord and others in the ^{my}Grid project for their enthusiasm, encouragement, and explicit and tacit support of our efforts. H.S., R.E., and D.H. thank all PlaNet partners, in particular, for making data and services available.

Received December 31, 2004; returned for revision March 3, 2005; accepted March 3, 2005.

LITERATURE CITED

Ashburner M, Ball CA, Blake JA (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet **25:** 25–29

Chicurel M (2002) Bioinformatics: bringing it all together. Nature **419:** 751–757

Durbin R, Thierry Mieg J (1991) A *C. elegans* database. http://www.acedb.org

Emanuelsson O, Nielsen H, Brunak S, von Heijne G (2000) Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. J Mol Biol **300:** 1005–1016

Fielding R (2000) Architectural styles and the design of network-based software architectures. PhD thesis. University of California, Irvine, CA

Goble C, Pettifer S, Stevens R, Greenhalgh C (2003) Knowledge integration: in silico experiments in bioinformatics. *In* I Foster, C Kesselman, eds, The Grid: Blueprint for a New Computing Infrastructure, Ed 2. Morgan Kaufman, San Francisco, Chapter 13

Gribskov M (2003) Challenges in data management for functional genomics. OMICS **7:** 3–5

Hernandez T, Kambhampati S (2004) Integration of biological sources: current systems and challenges ahead. *In* ACM SIGMOD Record, Volume 33, Issue 3. ACM Press, New York, pp 51–60

Karp P (1995) A strategy for database interoperation. J Comput Bio **2:** 573–583

Lewis SE (2004) Gene ontology: looking backwards and forwards. Genome Biol **6:** 103

Lewis SE, Searle SMJ, Harris N, Gibson M, Iyer V, Richter J, Wiel C, Bayraktaroglu L, Birney E, Crosby MA, et al (2002) Apollo: a sequence annotation editor. Genome Biol **3:** RESEARCH0082

Lord P, Alper P, Wroe C, Goble C (2005) Feta: a light-weight architecture for user-oriented semantic service discovery. *In* Proceedings of the European Semantic Web Conference (in press)

Lord P, Bechhofer S, Wilkinson M, Schiltz G, Gessler D, Hull D, Goble C, Stein L (2004) Applying semantic web services to bioinformatics: experiences gained, lessons learnt. *In* ISWC 2004. Springer Verlag, Berlin, 350–364

Louis A, Ollivier E, Aude JC, Risler JL (2001) Massive sequence comparisons as a help in annotating genomic sequences. Genome Res **11:** 1296–1303

Mohseni-Zadeh S, Louis A, Brézellec P, Risler JL (2004) PHYTOPROT: a database of clusters of plant proteins. Nucleic Acids Res (Database issue) **32:** 351–353

Oinn T, Addis M, Ferris J, Marvin D, Greenwood M, Carver T, Pocock MR, Wipat A, Li P (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics **20:** 3045–3054

Rhee SY, Beavis W, Berardini TZ, Chen G, Dixon D, Doyle A, Garcia-Hernandez M, Huala E, Lander G, Montoya M, et al (2003) The

Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. Nucleic Acids Res **31:** 224–228

Schoof H, Ernst R, Mayer KFX (2004) The PlaNet consortium: a network of European plant databases connecting plant genome data in an integrated biological knowledge resource. Comp Funct Genom **5:** 184–189

Schoof H, Zaccaria P, Gundlach H, Lemcke K, Rudd S, Kolesov G, Arnold R, Mewes HW, Mayer KFX (2002) MIPS Arabidopsis thaliana Database (MAtDB): an integrated biological knowledge resource based on the first complete plant genome. Nucleic Acids Res **30:** 91–93

Stein L (2002) Creating a bionformatics nation. Nature **417:** 119–120

Stein L (2003) Integrating biological databases. Nat Rev Genet **4:** 337–345

Stein L, Mungall C, Shu S, Caudy M, Mangone M, Day A, Nickerson E, Stajich J, Harris T, Arva A, et al (2002) The Generic Genome Browser: a building block for a model organism system database. Genome Res **12:** 1599–1610

Stevens R, Greenwood M, Goble CA (2003) Provenance of e-Science Experiments: experience from bioinformatics. *In* Proceedings UK OST e-Science Second All Hands Meeting 2003, Nottingham, UK, September 2–4, 2003

Stevens RD, Tipney HJ, Wroe CJ, Oinn TM, Senger M, Lord PW, Goble CA, Brass A, Tassabehji M (2004) Exploring Williams Beuren Syndrome using myGrid. Bioinformatics (Suppl 1) **20:** i303–i310

Wilkinson (2003) Gbrowse_moby: an integrated browser for MOBY-S web services. http://mobycentral.cbr.nrc.ca

Wilkinson M (2004) BioMOBY: the MOBY-S platform for interoperable data service provision. *In* RP Grant, ed, Computational Genomics. Horizon Bioscience, Wymondham, UK

Wilkinson MD, Gessler D, Farmer A, Stein L (2003) The BioMOBY project explores open-source, simple, extensible protocols for enabling biological database interoperability. *In* Proceeding of the Virtual Conference on Genomics and Bioinformatics, September 16–19, 2003, pp 17–27

Wilkinson MD, Links M (2002) BioMOBY: an open-source biological web services proposal. Brief Bioinform **3:** 331–341