Applying Semantic Web Services to Bioinformatics: Experiences Gained, Lessons Learnt

Phillip Lord[1], Sean Bechhofer[1], Mark D. Wilkinson[2], Gary Schiltz[3], Damian Gessler[3], Duncan Hull[1], Carole Goble[1], and Lincoln Stein[4]

[1] Department of Computer Science, University of Manchester
Oxford Road, Manchester, M13 9PL, UK
[2] University of British Columbia, James Hogg iCAPTURE Centre
St. Paul's Hospital, 1081 Burrard St., Vancouver, BC, V6Z 1Y3, Canada
[3] National Center for Genome Resources
2935 Rodeo Park Drive East, Santa Fe, NM 87505, US
[4] Cold Spring Harbor Laboratory
1 Bungtown Road, Cold Spring Harbor, NY 11724, US

Abstract. We have seen an increasing amount of interest in the application of Semantic Web technologies to Web services. The aim is to support automated discovery and composition of the services allowing seamless and transparent interoperability. In this paper we discuss three projects that are applying such technologies to bioinformatics: ^{my}Grid, MOBY-Services and Semantic-MOBY. Through an examination of the differences and similarities between the solutions produced, we highlight some of the practical difficulties in developing Semantic Web services and suggest that the experiences with these projects have implications for the development of Semantic Web services as a whole.

1 Introduction

In the past 10 years, the ability to perform biological *in silico* experiments has increased massively, largely due to the advent of high-throughput technologies that have enabled the industrialisation of data gathering.

There are two principal problems facing biological scientists in their desire to perform experiments with these data. The first of these is distribution–many of the data sets have been generated by individual groups around the world, and they control their data sets in an autonomous fashion. Secondly, biology is a highly heterogeneous field. There are large numbers of data types and of tools operating on these data types. Integration of these tools is difficult but vital. [2]

Biology has coped with this in an effective and yet very *ad hoc* manner. Almost all of the databases and tools of bioinformatics have been made available on the Web; the browser becoming an essential tool of the experimental biologist. The reasons for this choice of technology are partly chance in that the growth in genomic technologies happened to occur contemporaneously with the growth

S.A. McIlraith et al. (Eds.): ISWC 2004, LNCS 3298, pp. 350–364, 2004.

of the Web. But many of the key benefits of the Web are also important for biologists. Publishing is economically cheap, technically straightforward, innately distributed, decentralised, and resilient to change. Accessing the Web is likewise simple, requiring no knowledge of specific query languages but enabling "query by navigation" [6].

While this has worked well in the past, it has obvious problems. Many bioinformatics analyses use fragile screen-scraping technologies to access data. Keeping aware of the Web sites on offer is, in itself, a full-time and highly skilled task, mostly because of the complexity of the domain. The application of Semantic Web services to bioinformatics seems a sensible idea as Web services provide a programmatic interface which avoids screen-scraping [14], while semantic descriptions could enable their discovery and composition.

In this paper we describe three architectures, ^{my}Grid, MOBY-Services and Semantic-MOBY, which have been designed to address these problems. All three are aimed mainly at bioinformatics. All three are based on Web or Web-services technologies and use an additional specification of their services to describe the semantics of their operations. All three are high-profile projects in the domain of bioinformatics and come from groups with previous track records of providing solutions for problems of interoperability[1].

The three solutions are also different from each other and from the "idealised" Semantic Web services architecture. In examining these differences, we raise a set of key questions about the applicability of Semantic Web services in practice and present our (partial) solutions for these difficulties.

2 A Day in the Life: Bioinformatics as It Is

Bioinformatics as a discipline has largely grown directly out of the molecular-biology laboratories where it was born. In general, each lab investigated a small region of biology and there are very few labs world-wide working on a single problem. Many of these labs have made their own data available for use on the Web. This data is often un- or semi-structured. Much of the data is composed of DNA or protein sequences, but this has generally been accompanied by large quantities of "annotation"–descriptions (generally in free-text form) of the sources of the sequences, literature citations and the possible function(s) of the molecules. In addition to this raw data, many different tools that operate on it have been developed, most of them with restricted functionality and targeted at performing highly specific tasks.

This situation is slowly changing, largely due to the appearance of large **service providers**, such as the genome-sequencing and -annotating centres. These centres are now increasing their scopes and often provide many different types of information. The primary **service consumer** still remains the small laboratory. Much of the information remains openly accessible.

[1] To our knowledge, at the time of writing these were the only substantial projects using Semantic Web Services within bioinformatics

The primary "integration layer" so far has been expert biologists and bioinformaticians. Using their expert knowledge of the domain, they will navigate through the various Web pages offering data or tool access. Information about new resources often comes by word of mouth, through Web portals or paper publications. Data transfer, between applications, is by cut and paste, often with additional data "massaging" (*e.g.*, small alterations in formatting, selections of subsets, simple local transformations such as DNA-to-protein translation). Automation of these processes is achieved largely by bespoke code, often screen-scraping the same Web pages that the manual process would use, sometimes using more programmatically amenable forms of access.

From this description we identify the following:-

- Actors
  - **Service Providers.** Generally, but not exclusively, from specialised "genome" centres.
  - **Service Consumers.** Generally from smaller laboratories, normally with smaller, non-specialist resources.
- Requirements for integration
  - **Discovery and Description.** Finding the right data or tool resources is a complex task. Service providers need to be able to describe their services, and consumers discover services by these descriptions.
  - **Remote Programmatic Access.** Current screen-scraping technologies are fragile. Organised and preferably uniform access is required.
  - **Message Formatting.** Bespoke data massaging is complex and difficult to automate.

3 Semantic Web Services in a Nutshell

The core task of this a generic semantic web architecture is to enable seamless and inter-operable communication between service providers and service consumers.

It achieves this end with five key components:

Service Interfaces. Service providers publish interfaces to their services using some form of programmatic access.

Semantic Descriptions. In addition to the interface description, semantic descriptions of services are provided. OWL-S is the most prominent framework for these descriptions.

A Domain Ontology. Terms from an ontology describing the key concepts in the domain are used within the semantic descriptions.

Registry/Matchmaker. A matchmaker service searches over the semantic descriptions made available to it. This may be combined with a registry, a service which advertises the availability of other services.

Messaging. The domain ontology is used as a *lingua franca* that enables the service consumer to treat data from different providers in a uniform fashion.

Probably the best known work on supporting such architectures is OWL-S, following on from DAML-S [18]. OWL-S is an upper ontology that describes three key aspects about a service: its *profile*, which describes what the service does; its *process*, which describes how one interacts with the service; and its *grounding*, which relates the ontological concepts to the implementation, usually via a mapping to the WSDL operations.

In this paper we compare the architectures of the three projects with that of an idealised "Semantic Web Services architecture". All three of the projects are attempting to fulfill the requirements specified in Section 2 and are building components that aim to fulfill the role of the five key components described above.

From this comparison we draw the following conclusions:

- The importance of fully automated service discovery and composition is an open question. It is unclear whether it is either possible or desirable, for all services, in this domain, and is an area of research [22].
- Requiring service providers and consumers to re-structure their data in a new formalism for external integration is also inappropriate. External formalisms that adapt to the existence of legacy structuring is sufficient for many purposes.
- The service interfaces within bioinformatics are relatively simple. An extensible or constrained interoperability framework is likely to suffice for current demands: a fully generic framework is currently not necessary.
- If service discovery is to serve the user, descriptions based on users' own models of services are needed. Furthermore, contextual, outcome or task-oriented descriptions are required.
- Semantic services require a domain ontology, but the best way to construct one is not clear. We present three potential solutions for this problem.

4 The Projects

In this section, we present a brief introduction to the three projects. We then give a description of how a traditional "Semantic Web Services" architecture might be applied and explain how this has been implemented in the three projects.

The ^{my}Grid project is part of the UK government's e-Science programme [17]. It is aimed at providing open-source, high-level middleware to support personalised *in silico* experiments in biology. Although still at a prototype stage, ^{my}Grid has been used for two case studies. These have operated as focal points for the technology based around two diseases namely Graves' Disease [12] and Williams-Beuren syndrome [15]. The core ^{my}Grid "philosophy" has been to adopt Web services standards wherever possible and build additional middleware to add value to these. The key components described in Section 3 are realised within ^{my}Grid as follows:

Service Interfaces. Services are published as Web services described with WSDL.

Semantic Descriptions. A lightweight RDF data model is used to structure a service description, with a domain ontology providing a vocabulary. Descriptions can be provided by third parties. Previously ^{my}Grid used full DAML+OIL descriptions.

Domain Ontology. The ontology is curated and stored centrally, and generated by an expert using DAML+OIL.

Registry/Matchmaker. A centralised UDDI registry built over a Jena back end, augmented to enable semantic discovery [7].

Messaging. Pre-existing domain formats are used.

The Bio-Moby project[2] has grown from the "model organism" communities– those supporting the investigation of biological problems in different organisms. These communities have evolved standards which are often specific to one community. However, biologists increasingly wish to ask questions requiring data gathered from many different organisms, thus creating a severe integration problem. The Bio-Moby project has a dual development track with different architectures. The first of these, MOBY-Services (also known as "MOBY-S"), has "simplicity and familiarity" as its core philosophy. MOBY-Services [21] exists as a prototype that is in practical use at a number of sites. The key components are realised within MOBY-Services as follows:

Service Interfaces. Services are simplified compared to WSDL, having single operations, inputs and outputs.

Semantic Descriptions. A data model is enforced by the API of registry with a domain ontology providing a vocabulary.

Domain Ontology. The ontology is user curated, stored centrally, generate by community collaboration, and structure as a Gene Ontology style DAG.

Registry/Matchmaker. A centralised bespoke registry called "MOBY-Central", which enables searching by input and output types, augmented with graph crawling.

Messaging. A thin XML envelope with embedded legacy formats is used.

The second of these, Semantic-MOBY [20] (also known as S-MOBY), has been heavily influenced by the REST architectural style [3] and makes extensive use of Semantic Web technology, in particular OWL-DL. It attempts to embrace the autonomous nature of the Web wherever possible. Also at a prototype stage, Semantic-MOBY has extensive publicly available requirements[3] and design[4] documentation. The key components are realised within Semantic-MOBY as follows:

Service Interfaces. Services are simply Web resources accessible by standard protocols such as HTTP and FTP. For example, via HTTP, a simple GET returns an RDF graph that defines the underlying service interface.

[2] `http://www.biomoby.org`

[3] `http://www.biomoby.org/S-MOBY/doc/Requirements/S-MOBY_Requirements.pdf`

[4] `http://www.biomoby.org/S-MOBY/doc/Design/S-MOBY_Design.pdf`

Semantic Descriptions. Service descriptions are expressed in OWL-DL and conform to a canonical format, or upper ontology. This upper ontology creates the context for ontological concepts, which are resolvable into OWL-DL graphs themselves by dereferencing their URIs. Service providers create service-specific subclasses of the ontology, grounding them with their own data-type requirements.

Domain Ontology. One, or several, ontologies are developed by the community, and distributed across the Web, and written in OWL-DL.

Matchmaker. One or more centralised search engines are provided. Service locations can be published, or semantic descriptions can be discovered by Web crawlers. Querying uses the same upper ontology as the semantic descriptions.

Messaging. All communication uses OWL-DL and the same upper ontology.

All three projects therefore share a strong biological focus. They are interested in easing the difficulty of connecting existing service providers to existing service consumers within this domain. This is more important than providing a generic solution.

5 Automated Service Composition

As a domain, bioinformatics has a number of characteristics of relevance to automated service composition.

Complexity. It is unlikely that any representation of the domain and background knowledge will come close to matching the knowledge of the expert bioinformatician.

Fluidity. Key concepts in the domain are open to change. Any codified representation of the domain is likely to be out of date.

Diversity. Opinions differ. Bioinformaticians wish to be involved in the selection of services to ensure that their opinions are reflected.

Automated composition is likely to be useful where transparent seamless access is the most overriding requirement, such as booking appointments [23]. Here, users will be happy to accept the results, so long as they are reasonable and they gain the advantage of not having to perform such tasks themselves. It is not likely to serve the needs of expert, knowledgeable, opinionated scientists who may invest large quantities of money and time in further experiments based on the results and who may be required to justify their methodologies under peer review. In the short term, these scientists are unlikely to trust automated service invocation and composition, probably with justification, as it is unlikely to improve on their own selections. We wish to support biologists' activities, not replace them. In this way, bioinformatics seems to be following the path of medical informatics, where early decision-making systems have given way to later decision-support systems [9].

Our requirements analyses showed one exception to this: the selection of one service from a collection of mirrors. Like most computer users, biologists are fond of asking for ways to "make it go faster". However, a set of services that are mirrors (particularly databases) must have a coordinated update strategy, probably on a daily basis. This coordination indicates that they are not truly autonomous. They are also likely to share the same user and service interfaces (and probably the same code base), so there is no heterogeneity between the mirrors either. Given this, it seems that semantic descriptions are unlikely to be useful in choosing between them. The *my*Grid project is also investigating other services, for which automated discovery and composition may be useful; these are described in Section 6.

6 Structured Messages and Middleware

Combined with the complexity of biology, the autonomous nature of bioinformatics has made integration of the different data resources extremely difficult. Most of the key data types, in bioinformatics, have no standard representation or many "standard representations"; there are at least 20 different formats for representing DNA sequences, most of which have no formal specification[5]. As a simple two-bit code, a DNA sequence is at its essence one of the simplest biological data types, and there are many data types which are considerably more complex.

Where standards do exist, they have often arisen as a result of many years of collaborative work. Both the service providers and service consumers have a high degree of buy-in to the formalisms that exist. The service consumers want their data in legacy formats because their tools can operate over them. While some of the data types are simple, many, however, are highly complex and internally structured[6]

Previous work has highlighted this difficulty with web services from other domains. Paolucci et al. [11], note that "the [complex] types used in a WSDL specification [for accessing amazon.com] are totally arbitrary". This problem is taken to an extreme in bioinformatics in that complex types are simply not used. To demonstrate this point, we gathered 30 bioinformatics service descriptions[7]. Of these only two defined any complex types at all and one of these was a simple list type. Bioinformaticians are just not structuring their data in XML schema, because it provides little value to them. All three projects have accepted that much of the data that they receive will not be structured in a standard way. The obvious corollary of this is that without restructuring, the information will be largely opaque to the service layer.

[5] `http://www.hgmp.mrc.ac.uk/Software/EMBOSS/Themes/SequenceFormats.html`

[6] The Swissprot database, for instance, has a 30 pages human readable specification for their format. Alternatively a regexp grammar designed to parse the database is about 300 lines long, and it does not fully parse all the implicit structure

[7] `http://www.ebi.ac.uk/~tmo/mygrid/webservices/`

The projects have coped with this in different ways. The simplest approach is that of ^{my}Grid. Service providers are largely autonomous and are unlikely to change their data formats unless there are compelling reasons. ^{my}Grid therefore imposes no additional structuring on the data. While there are problems stemming from the lack of a formal representation of many of the formats, bioinformatics has had a long time to solve these problems; there are many good tools and parsers that are capable of coping with this multitude of formats. Within the ^{my}Grid project, the existence of these formats is described; this then should enable us to discover services that can translate between these formats, even though these formats are opaque to the middleware. This process has been previously described as *automatic semantic translation* [8]. We describe these services as "shim services"[8]. In addition to format translation there are several other situations where users can be supported in composing services by the discovery of shims. Firstly, identifier dereferencing–bioinformatics makes extensive use of identifiers, and substituting an identifier for the data it describes is a common task [16]; and secondly, decomposition–selecting a subset of the data produced by one service, for use by another.

In these cases, it appears that services can be automatically composed in a "safe" manner, *i.e.* that is, they do not change the intended meaning of the experimental design, but enable it to work.

Both MOBY-Services and Semantic-MOBY do provide additional structuring for their messaging format. MOBY-Services uses a specialised XML schema, that defines all of its messages, which mostly acts as a thin envelope around the opaque structuring of the legacy formats. However, having accepted the existence of a wrapper, MOBY-Services now has a migration path to the increasing structuring of data. Currently it has used this facility to introduce "Cross-References"; one of the most characteristic features of the bioinformatics data has been the addition of links to other "related" data encoded mainly as hyperlinks to other databases. The cross-references provide a similar facility for programmatically accessed services.

Semantic-MOBY on the other hand uses OWL-DL RDF-XML as a messaging layer and OWL-DL for content structuring. Ontological concepts are mapped to XSD data types by providers in the description of their resource. The rich expressivity of OWL means that, over time, concept descriptions can be customized by extending domain ontologies with new properties, combining concepts across ontologies, and constraining usage with OWL-DL property restrictions.

In short, while none of the projects assume that providers will automatically restructure their data into formats defined in XML Schemas, they all provide migration paths to methodologies that can reduce the problem of syntactic heterogeneity.

[8] A shim is a small piece of material used to fill gaps to ensure a tight fit or level a surface

7 Service Provision and Service Interfaces

The key sociological problem that any interoperability middleware faces is attempting to gain support from the service providers. They are semi-autonomous tending to respond well to requests from service consumers. This is unlikely to happen until the system is usable, but a usable system requires the existence of many services.

The MOBY-Services project has taken a different approach. It assumes that service providers are more likely to support MOBY-Services-specific service interfaces if they are easy to generate and require minimal changes to the way the service providers already work. Within the realm of bioinformatics, most existing services look somewhat like a command-line application. Each has one input and one output. There are usually a set of parameters, *i.e.*, values which modify the way the service works, much like command-line options. We describe this style of interface in Table 1 as "Document Style". MOBY-Services therefore uses a subset of the full Web services functionality, thus limiting service interfaces to this basic data model. Each service is atomic, stateless and unrelated to other services.

The ^{my}Grid project assumes that various service providers will develop Web service interfaces for their own purposes and the middleware solutions should just be able to cope with these. Additionally, it has developed a set of services of its own with a package called "SOAPLAB" [13]. Interestingly, most of the available services conform to one of two paradigms. Most of the services have operations which conform to the simplified model of MOBY-Services: inputs, outputs, parameters. Many service providers group related, although independent, operations into a single service; MOBY-Services services only ever have a single operation. The second paradigm, described in Table 1 as "Object Style", is used by SOAPLAB[9]. The service operations define an *ad hoc* object model. This creates a problem in terms of invocation. Any client that uses these services must understand the semantics of the operations. Within the ^{my}Grid project, this has been enabled by using an extensible invocation framework within "freefluo": the workflow-enactment engine [10]. This approach falls short of a generic solution in that the framework needs to be extended for different "styles" of Web services. But within the constrained domain of bioinformatics, with its relatively few service providers, it is likely to be sufficient.

The Semantic-MOBY service interface is somewhat different. Following the REST architecture, each service has an interface based on, for example, HTTP. The underlying messages provide a richer interface defined by the Semantic-MOBY upper ontology. Like MOBY-Services, this service interface is atomic, stateless, and unrelated to other services.

Compared to general WSDL services, the simplified approach that all three projects have taken to service interfaces has some important consequences.

[9] It has been argued that web services are not designed to support such object style interfaces [19]. This may be true; however if ^{my}Grid aims to use services from autonomous providers it needs to cope with them.

Table 1. Two different service interfaces to BLAST, a widely used bioinformatics tool. BLAST operates over a biological sequence, has a number of parameters and returns a single complex BLAST report. The "Document Style" interface has a single method taking a complex set of parameters, while the "Object Style" interface uses object identifiers to provide an *ad hoc* object orientation.

Document Style	BlastReport performBlast(Sequence, gap, etc. . .);
Object Style	ObjectIdentifier getInstance();
	void setSequence(ObjectIdentifier, Sequence);
	void setGap(ObjectIdentifier, Gap);
	. . .
	BlastReport invoke(ObjectIdentifier);

OWL-S defines a *grounding* ontology, which describes the relationships between the ontological concepts in the underlying invocation description (*e.g.*, WSDL). This is not required for any of the projects: Semantic-MOBY because its "service interface" is defined by its upper ontology; MOBY-Services because its service interfaces are not heterogeneous; and, ^{my}Grid because the enactment engine deals with the small amount of heterogeneity.

The services are all atomic and not decomposable. As a result, nothing similar to the OWL-S *process* ontology has been used. Only ^{my}Grid uses services that require complex interaction (namely the SOAPLAB services), and this interaction is handled by the enactment engine.

8 User-Centred Service Descriptions

A Semantic Web Services architecture requires a set of semantic service descriptions, which a matchmaker service can then use to discover services.

Within these three projects, all are seeking to enable discovery of services by the user. All three projects share similar ideas about the questions that users wish to ask.

Context. The user has a specific piece of data and wishes to know which services can operate on that type of data.

Outcome. The user wishes to get a specific type of data and wishes to know which services can produce this kind of data.

Task. The user knows what kind of task, *e.g. alignment, retrieval or search.*, to perform, and wishes to known how.

The ^{my}Grid project has introduced a number of other properties of services, including one called "uses resource". Many bioinformatics tools can operate over different data sources. The underlying data has a critical impact on the use of a service, but may well not affect the interface it presents.

In a user-centred approach the users' concepts of "services" do not necessarily confirm to that of the underlying middleware. For example, within ^{my}Grid,

services are generally a collection of independent, but related, operations. However, with object-style interfaces, as described in Section 7, the user wishes to find the whole service; the individual operations are essentially an implementation detail. With MOBY-Services, this abstraction is directly represented in the middleware in that each service has only one operation. A second example of this is the distinction between *inputs* and *parameters*[10]. This distinction can be understood by analogy to a command line which has a main argument and a set of options, or switches. In general, users are interested in searching for inputs, while parameters can be safely ignored until invocation time. As well as serving the users' needs, this distinction has the serendipitous effect of greatly reducing the space over which any technology must search.

The biggest limitation of service descriptions at the moment is their assumption of a single class of user. For example, the ^{my}Grid ontology is aimed at bioinformaticians, as this seemed the main user base. Previous systems, such as TAMBIS [4], have been more aimed at biologists; hence biological concepts such as "protein" are modelled and presented instead of bioinformatics concepts such as "Swissprot ID".

The three projects have somewhat differing ideas about who will provide service descriptions. Both MOBY-Services and Semantic-MOBY have systems based on service providers who describe their own services. Conversely, as with its approach to service provision, the ^{my}Grid project has concluded that service providers may or may not choose to do so, and that, at least in the short term, descriptions by third-party members (*e.g.*, those within the ^{my}Grid project!) are essential, and may be desirable even in the long term.

Service descriptions are currently manually generated by all three projects. The relatively simple and familiar formalism used by MOBY-Services (see Section 9) ensures that this process is relatively straightforward. Alternatively, Semantic-MOBY and ^{my}Grid are relying on the provision of good tool support to ease the process. The ^{my}Grid project has made perhaps the most effort in this area reusing a tool from the PEDRo project[11] which presents the user with a fill in form, and then generates services descriptions from this. Describing services is still, however, an arduous process–even though the simplified service interfaces reduce the required complexity of the service descriptions (Section 7). In addition the use of legacy formats (Section 6), means very little information can be mined from the WSDL files; tools such as WSDL2DAML-S are relatively ineffective when they have no information to work on. Most information has been gained from reading associated documentation, guesswork based on service/operation names and experimental execution of the services followed by manual inspection of the results[12].

It is clear that these approaches are not scalable. It is possible that more automated techniques [5] may become applicable in the future. Within the re-

[10] *primary* and *secondary* inputs in MOBY-Services parlance

[11] `http://pedro.man.ac.uk/`

[12] That it is so difficult to work out what services actually do, demonstrates clearly that better service descriptions are genuinely needed!

stricted domain of bioinformatics, it should be possible to partially automate experimental service execution. In the short term, however, the authoring of service descriptions will remain a major bottleneck.

9 Generating an Ontology for a Complex Domain

One fundamental difficulty of any Semantic Web Services architectures is the requirement for a domain ontology. The ontology must reflect the users' understanding of the domain and enable the description of services by service providers or helpful third parties, particularly in support of user-oriented service discovery. In a complex and changing domain such as bioinformatics this is no small undertaking.

Fortunately, within bioinformatics the community is already conversant and are generally convinced of the value of ontological descriptions. Over the last four years, the community has developed the *Gene Ontology* (GO). This is a hierarchically organised controlled vocabulary. The ontology is structurally simple; having subsumption (*is-a*) and partonomy (*part-of*) hierarchies. The ontology is widely used; it now consists of ∼17,000 terms and has been used to describe several million database entries. GO is the flagship ontology of a collection of biological ontologies, called OBO (Open Biological Ontologies), which describe various aspects of biology.

GO has provided much of the inspiration for the MOBY-Services approach to ontology development. One of the biggest factors in the success of GO has been the large level of community involvement in its construction[1]. The MOBY-Services central registry *MOBY-Central* therefore contains functionality for adding new terms and relationships. It has adopted the same representational formalism as GO as this is already familiar to the community; familiarity is considered more important than expressivity. Combined with reasonable editorial guidelines and an active mailing list it is hoped that those using the ontology will extend it to fulfil their requirements. Additionally, MOBY-Services has already constructed tools for viewing their ontology; it is essential that finding existing appropriate concepts is easy, in order to discourage the recreation of existing concepts. We call this the *collaborative community* style of ontology building.

^{my}Grid has learnt a different lesson from the GO, namely the importance of the role of a curator. To this end a large ontology has been already been constructed by the project. In this case, ^{my}Grid chose to use DAML+OIL as its underlying formalism because of the existence of tooling (*e.g.*, OilEd) and reasoning capabilities. Wherever possible, existing classifications from the community were used. By analysing use cases from the other two projects that were unavailable at the time the ontology was constructed, it is clear that it has reasonable, but incomplete, coverage of the domain. It is clear that, like MOBY-Services, methods to encourage feedback from the community are essential. We call this *centralised, curated* ontology building.

These two methods are not exclusive, however. MOBY-Services and ^{my}Grid are therefore making significant efforts to align their ontologies. As the largest

currently available ontology of bioinformatics services, it is hoped that a view of the ^{my}Grid ontology can be constructed so that it can be used with the MOBY-Services project. The collaborative community would then be aiding knowledge capture rather than ontology building *per se*. This is perhaps closest to the Gene Ontology process. It still, however, suffers from the problem that the cost of curating the ontology must be borne centrally rather than distributed through the community.

Finally, Semantic-MOBY has sought to embrace a *distributed, autonomous* style of ontology building, reflecting the nature of bioinformatics. It seems likely that, over time, the community will build ontologies describing some or all of the domain. The independent development of the ^{my}Grid ontology seems to prove that this assumption is accurate. The Semantic-MOBY infrastructure has been designed to cope with a free market of ontology development. However, ontologies provide interoperability only so far as they are shared by members of the community. If multiple ontologies are allowed or encouraged yet mechanisms for interoperability are not embedded, there is a risk that independent "city states" will develop.

The use of a highly compositional style of ontology development, backed by a well-defined semantics and inference is critical to avoiding this development. It is hoped that core concepts can be created with community agreement and that these concepts can be extended by smaller specialist groups. This approach is also supported by the existence of the other projects. Any ontology developed by either or both projects that is reflective of community opinion will be likely to find favour within Semantic-MOBY's free market.

Of all the aspects of the three architectures, ontology building is where the three differ the most. Perhaps this stems from the fundamental difficulty of the issue.

Given these differences, it is ironic, although heartening, that ontology building is also the activity where all three projects have shown the highest degree of collaboration, engaging in active efforts to align semantics, share the knowledge already represented explicitly, and gather additional knowledge implicit within the community.

10 Discussion

Semantic Web Services technologies offer the prospect of increased interoperability and of (semi)automated service discovery and invocation. The advent of high-throughput biological techniques has made the requirement for such a technology within bioinformatics immediate and pressing.

In this paper we have described three architectures that are applying these technologies. We have also described some of the large practical problems that have presented themselves.

Clearly all three projects are aimed at supporting bioinformatics. We feel, however, that some of the experiences may be relevant to other domains. We suggest:-

- The inappropriateness of automated service invocation and composition is likely to be found in other scientific or highly technical domains.
- The difficulties in providing a domain ontology are likely to be shared by any domain with complex data types; structuring of data is likely to be difficult where significant legacy systems exist. We suspect that our solutions may be practical for use only within a restricted domain.
- Semantic Service Discovery tailored toward the users' notions of services is likely to be a useful augmentation to all domains.

At the current time, we are unsure as to the applicability of simple service interfaces to other domains, although we suspect this may remain relatively unique to bioinformatics.

Despite these difficulties it seems likely that the development of Semantic Web Services technologies should ease the difficulty of service discovery and composition within bioinformatics. Conversely, we believe that bioinformatics offers good opportunities for testing these technologies in a practical setting.

Acknowledgements. PL and DH are supported by the UK e-Science programme EPSRC grant GR/R67743 as part of the ^{my}Grid project. SB was supported by the WonderWeb project (EU grant IST-2001-33052). DG, GS and LS thank Andrew Farmer, Ardavan Kanani, Fiona Cunningham and Shuly Avraham for their contributions to Semantic-MOBY. Semantic-MOBY is funded by a National Science Foundation grant 0213512 to L.S. and D.G. The MOBY-Services project wishes to thank Dr. William Crosby, Mr. Matthew Links, Mr. Luke McCarthy, and the National Research Council Canada for their advice, intellectual, and financial contributions. The work on MOBY-Services was supported by a grant from Genome Canada/Genome Prairie to MDW.

References

1. Michael Bada, Robert Stevens, Carole Goble, Yolanda Gil, Michael Ashburner, Judith A. Blake4, J. Michael Cherry, Midori Harris, and Suzanna Lewis. A Short Study on the Success of the Gene Ontology. Accepted for publication in the Journal of Web Semantics, 2004.
2. Marina Chicurel. Bioinformatics: bringing it all together. *Nature*, 419:751–757, 2002.
3. Roy T. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.
4. C.A. Goble, R. Stevens, G. Ng, S. Bechhofer, N.W. Paton, P.G. Baker, M. Peim, and A. Brass. Transparent Access to Multiple Bioinformatics Information Sources. *IBM Systems Journal Special issue on deep computing for the life sciences*, 40(2):532 – 552, 2001.
5. Andreas Heß and Nicholas Kushmerick. Learning to attach semantic metadata to web services. In *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*, Sanibel Island, Florida, October 2003.
6. P. Karp. A Strategy for Database Interoperation. *Journal of Computational Biology*, 2(4):573–586, 1995.

7. Phillip Lord, Chris Wroe, Robert Stevens, Carole Goble, Simon Miles, Luc Moreau, Keith Decker, Terry Payne, and Juri Papay. Semantic and Personalised Service Discovery. In W. K. Cheung and Y. Ye, editors, *WI/IAT 2003 Workshop on Knowledge Grid and Grid Intelligence*, pages 100–107, Halifax, Canada, October 2003.
8. Daniel J. Mandell and Sheila A. McIlraith. Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation. In *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*, Sanibel Island, Florida, October 2003.
9. R. A. Miller. Medical diagnostic decision support systems–past, present, and future: a threaded bibliography and brief commentary. *J Am Med Inform Assoc*, 1(1):8–27, Jan-Feb 1994.
10. T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Greenwood, T. Carver, A. Wipat, and P. Li. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 2004. Accepted for publication.
11. Massimo Paolucci, Anupriya Ankolekar, Naveen Srinivasan, and Katia Sycara. The DAML-S Virtual Machine. In *Internaional Semantic Web Conference*, pages 290 – 305, 2003.
12. Stevens R, K. Glover, C. Greenhalgh, C. Jennings, S. Pearce, P. Li, M. Radenkovic, and A. Wipat. Performing *in silico* Experiments on the Grid: A Users' Perspective. In *Proc UK e-Science programme All Hands Conference*, 2003.
13. Martin Senger, Peter Rice, and Tom Oinn. Soaplab – a unified sesame door to analysis tools. In *Proc UK e-Science programme All Hands Conference*, 2003.
14. Lincoln Stein. Creating a bioinformatics nation. *Nature*, 417:119 – 120, 2002.
15. R. Stevens, H.J. Tipney, C. Wroe andT. Oinn, M. Senger, C.A. Goble P. Lord, A. Brass, and M. Tassabehji. Exploring Williams-Beuren Syndrome using ^{my}Grid. In *Proceedings of 12th International Conference on Intelligent Systems in Molecular Biology*, 2004. Accepted for Publication.
16. R.D. Stevens, C.A. Goble, P. Baker, and A. Brass. A Classification of Tasks in Bioinformatics. *Bioinformatics*, 17(2):180–188, 2001.
17. Robert D. Stevens, Alan J. Robinson, and Carole A. Goble. ^{my}Grid: personalised bioinformatics on the information grid. *Bioinformatics*, 19(90001):302i–304, 2003.
18. Katia Sycara, Massimo Paolucci, Anupriya Ankolekar, and Naveen Srinivasan. Automated discovery, interaction and composition of semantic web services. *Journal of Web Semantics*, 1(1):27–46, September 2003.
19. Werner Vogels. Web services are not distributed objects. *IEEE Internet Computing*, 7(6):59–66, 2003.
20. M. D. Wilkinson, D. Gessler, A. Farmer, and L. Stein. The BioMOBY Project Explores Open-Source, Simple, Extensible Protocols for Enabling Biological Database Interoperability. *Proc Virt Conf Genom and Bioinf*, 3:16–26, 2003.
21. M. D. Wilkinson and M. Links. BioMOBY: an open-source biological web services proposal. *Briefings In Bioinformatics*, 3(4):331–341, 2002.
22. C. Wroe, C. Goble, M. Greenwood, P. Lord, S. Miles, J. Papay, T. Payne, and L. Moreau. Automating experiments using semantic data on a bioinformatics grid. *IEEE Intelligent Systems*, 19(1):48–55, 2004.
23. Dan Wu, Bijan Parsia, Evren Sirin, James Hendler, and Dana Nau. Automating DAML-S web services composition using SHOP2. In *Proceedings of 2nd International Semantic Web Conference (ISWC2003)*, Sanibel Island, Florida, October 2003.