Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

DRUG DISCOVERY TODAY BIOSILICO

Web services in the life sciences

Vasa Curcin, Moustafa Ghanem and Yike Guo

Web services provide a standard way of publishing applications and data sources over the internet, enabling mass dissemination of knowledge. In the life sciences, the web-service approach is seen as being a road to standardizing the multitude of tools available from different providers. In this article, we present an overview of the technology (focusing on life-science applications), we list the currently available service providers and we discuss advanced issues raised by the concept.

The World Wide Web changed the way that people perceive information and its dissemination. It has established itself as a *de facto* standard for information sharing between individuals across the globe. Guided by this example, there has been notable interest in providing a similar methodology for sharing software. Tools would become accessible, as remote services, over the web, and uniform and standardized mechanisms would be used for service lookup, invocation and composition. This service-oriented approach has one notable benefit in that it enables non-IT users to construct new custom applications by putting together various existing remote databases and programs: effectively, simplifying programming to the level of connecting components.

Within the bioinformatics community, in which an average end-user might need to access and use hundreds of databases and tools on a given day, the success of this approach is a dream come true. The idea resounded deeply and quickly. Organizations such as the European Bioinformatics Institute (EBI; http://www.ebi.ac.uk/), DNA Data Bank of Japan (DDBJ; http://www.ddbj.nig.ac.jp/) and Virginia Bioinformatics Institute (VBI; https://www.vbi.vt.edu/) published their tools as web services (Table 1). Software environments and tools emerged to enable the execution, visualization and management of data associated with such services – the most notable being Discovery Net (commercialized as InforSense KDE; http://www.inforsense.com/) [1], Taverna (part of the myGrid project; http://www.mygrid.org.uk/) [2] and BioMoby [3]. Standards for service composition, including the Business Process Execution Language for Web Services (BPEL4WS) and the Business Process Modelling Language (BPML), also appeared. Moreover, research into the use of Ontology Web Language Semantics (OWL-S) [4] and Web Service Modelling Ontology (WSMO) [5] was conducted to investigate the use of ontologies and semantic web [6] technologies in the provision of higher-level and domain-specific methods for describing, discovering and composing service-based applications.

Figure 1 shows a small workflow application built out of web services provided by EMBL (http://www.embl.org/) [through the Sequence Retrieval System (SRS) – see later for description] and Kyoto Encyclopedia of Genes and Genomes (KEGG; http://www.genome.jp/kegg/). A list of accession numbers is first passed through the EMBL web service to obtain identifier mappings; it is then passed to several service calls to KEGG resources, retrieving pathways for genes and other related information. The individual nodes in the workflow are web services and local data transformations that massage the data into the correct format before passing them on to the next service. The end result is a list of pathway images, with the

Vasa Curcin*
Moustafa Ghanem
Yike Guo
Department of Computing,
Imperial College,
180 Queen's Gate,
London,
UK, SW7 2AZ
*e-mail: vc100@doc.ic.ac.uk

1359-6446/05/$ – see front matter PII: S1359-6446(05)03481-1

TABLE 1

Currently available production-quality web services in the life sciences

Organization	WSDL	Description
EBI	http://www.ebi.ac.uk/xembl/XEMBL.wsdl	EMBL interface
	http://industry.ebi.ac.uk/openBQS/copies/BQSWebService.wsdl	Bibliography query system
	http://www.ebi.ac.uk/soaplab/wsdl/	Mostly EMBOSS tools created through Soaplab interface
GenomeNet	http://soap.genome.jp/KEGG.wsdl	KEGG and associated databases
DDBJ	http://xml.ddbj.nig.ac.jp/wsdl/index.jsp	BLAST, SRS, FASTA, ClustalW and others
VBI	http://staff.vbi.vt.edu/pathport/services/	PathPort project services
National Cancer Institute	http://cabio.nci.nih.gov/soap/services/index.html	Several services from a variety of domains

genes from the original list marked on top of the pathway map.

In this article, we provide an overview of the basics of web services, we describe recent developments in the field and we list some of the currently available life-science services that can be used today.

Web-service basics

The driving goal of web-service research is to create a framework in which applications distributed across the internet can interoperate through a set of standard protocols. This fundamentally simple and powerful vision was plagued from the outset by rival standards in industry and academia, and lack of agreement about what level of control the web services should provide over underlying applications. Eventually, three standards emerged that established the most basic framework for representing and manipulating web services: the Web Services Description Language (WSDL), Simple Object Access Protocol (SOAP), and Universal Discovery, Description and Integration (UDDI). WSDL is an Extensible Markup Language (XML)-based standard for describing web services and specifying their parameters, inputs and outputs*. SOAP maps the abstract services described in WSDL to their concrete implementations, describing in detail the objects that are passed to and from web services and enabling them to be executed by the remote client. UDDI enables service providers to publish their services for the community, and potential end-users of the services to browse and look up specific services (http://www.w3.org/). This discovery aspect removes the need for the user to know exactly which service is required and where it is located†. Detailed descriptions of the first two standards are available at the World Wide Web Consortium (W3C; http://www.w3.org/), which is one of the international-standards bodies overlooking the development of web-service technologies. UDDI is maintained and documented at the Organization for the Advancement of Structural Information Standards (OASIS; http://www.oasis-open.org/home/index.php), which is another web-technology-standards organization. Finally, the Web Service Interoperability Organization (WS-I; http://www.ws-i.org/) is dedicated to promoting guidelines that help different web-service standards work together. These guidelines take the form of recommendation documents that are augmented by example applications and testing tools that ensure conformance.

Despite the popular belief that the presence of SOAP, WSDL or UDDI in a web application makes it a web service, this is not strictly true according to the definition by W3C, which does not specify explicitly the protocols to be used, except that they must be XML based. The importance of this distinction is that there might be different standards for describing a service and its parameters, and discovering that service on the network.

Web services are, essentially, an implementation of the service-oriented computing paradigm. The basic model for service-oriented architectures is shown in Figure 2. 'Consumer' represents the user, or user's application, that wants to make use of the service. It connects to the service registry to obtain the information about resources providing that service. After it has done

*It is worth noting that WSDL can be used to describe any type of service, including legacy applications such as R (http://www.r-project.org/), which provides descriptors for each procedure call, or File Transfer Protocol (in which descriptors would cover get and put requests).

†However, UDDI is not a perfect solution to the discovery problem because it does not provide a way to query the service information beyond the most basic naming information. No associated metadata are stored with UDDI.

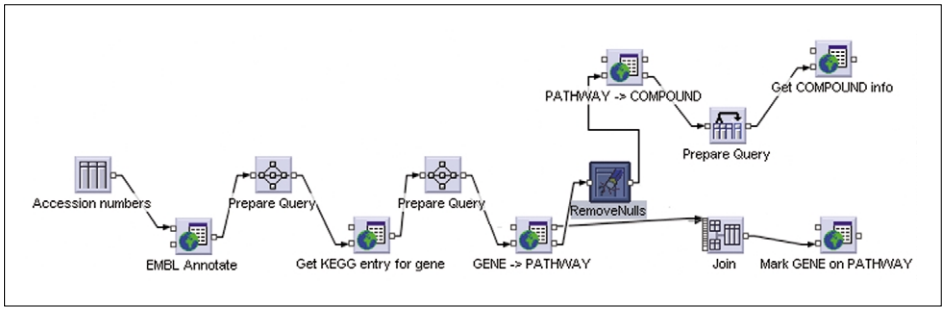

FIGURE 1

Mapping a list of accession numbers to KEGG pathways in InforSense service-composition software. Accession numbers are sent through the SRS web service to retrieve EMBL annotations that are then passed on to several KEGG web services.

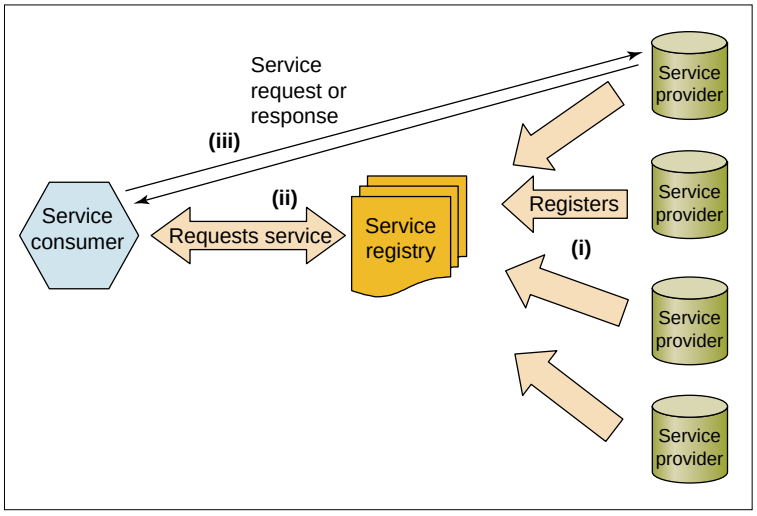

FIGURE 2

Overview of service-oriented architecture. (i) The service provider registers with the registry. **(ii)** The consumer requests the service from the registry. **(iii)** The consumer contacts the provider.

so, it contacts the resource directly with the appropriately formatted input data and retrieves the answer. For example, using the technologies listed earlier, SOAP would be used to describe inputs and outputs to the service, and the information about the service itself would be published using WSDL by the providers to the UDDI service registry and passed on to the consumers.

In parallel to the development of web services, several technologies appeared for mapping existing Java, Common Object Request Broker Architecture (CORBA) and Perl applications to and from web services (Box 1). These tools aim to help port the existing applications into web services by mapping types and method parameters into SOAP, and methods into WSDL.

Requirements for life-science web services

Following the period in which sequence informatics, metabonomics, cheminformatics and other areas were islands unto themselves, the life-sciences field is now increasingly focusing on building heterogeneous models obtained from different analysis domains and linking them to gain holistic understanding about how these domains interact with each other and where lie the 'blind spots' in the comprehension of a biological system. Here, the added value of web-service architectures is to enable different end-users, with different skill sets and tools, to create software components that can interact with one another.

Application integration

Combining different existing tools and applications to work together to perform analysis on the same dataset is not an idea unique to the web services. Piping Perl programs together is an established way of rapidly assembling software systems in bioinformatics. The downside to this approach is that it might pose long-term maintenance problems because it leads to vast amounts of multiple versions of scripts scattered around the file system. However, chained Perl scripts did establish two major paradigms in the integration of life-science applications (as documented in Ref. [7]): running multiple independent tasks in parallel (such as DNA annotation, identification of repeats and GpC islands) and performing multiple tasks in series (identifying homologues of a sequence; from these homologues, selecting either *n* closest matches or a manually selected set, aligning them and sending the output to various phylogenetic or dodistic packages).

In the latter model, there is an interaction between the applications involved. The results from one task must be included as an input to the next and, consequently, the output format of the first task must be synchronized with the input format of the second. The issue of format compatibility is one of the fundamental challenges of dynamic service composition.

Another notable characteristic here is the flexibility in adding components to the analysis process. New tools for each of these tasks are continually being developed by researchers and made available over the web (either as downloadable code or as remote services). Thus, in addition to process models and format resolution, a natural requirement after a good offering of web services is obtained is for a rapid integration mechanism that enables the inclusion of new tools and algorithms when they are needed by the researcher.

Heterogeneous data integration

The number and content of bioinformatics data sources are increasing. Data from primary databases that hold genomic sequences [e.g. EMBL, GenBank (http://www.ncbi.nlm.nih.gov/Genbank/) and LocusLink (http://www.ncbi.nlm.nih.gov/projects/LocusLink/)] are continuously being analysed, annotated and cross-referenced by human experts. The outputs are then deposited into new organism-specific, domain-specific or disease-specific sources. These sources differ not only in their format but also in their internal structure – the same biological objects can be organized based on

BOX 1

Precursors to web services. Web services are not the first software solution to the problem of distributed resource invocation. CORBA, Distributed Component Object Model (DCOM) and Remote Method Invocation (RMI) have all attempted to provide an extensible way for distributed applications to talk to each other. Even though these tools have had a major impact on large, enterprise-scale environments, they have several core problems that have made them unfeasible for open environments. First, owing to their proprietary nature, they could not provide effective cross-vendor communication – special CORBA–RMI–DCOM bridges had to be built, defeating the point of distributed computing. Second, all the data passed between hosts assumed that both client and server knew exactly what inputs and outputs to expect – there was no SOAP-style object description possible. Third, binary-encoded objects are far more difficult to decode and analyse than typically human-readable XML. The importance of web services is that they raise the distributed computing abstraction to a higher level, thereby removing a large part of intercommunication difficulties. Also, the open-source orientation of web services ensures that no single vendor can dictate their development and direction.

the pathways they belong to, their functional characteristics or the functional characteristics of the proteins they encode.

The SRS software [8], commercialized by Lion BioSciences (http://www.lionbioscience.com/), tackles this problem by introducing indices for each database that supports optimized lazy parsing (whereby the client requests that the parser be constructed dynamically for a specific query, thereby avoiding unnecessary processing, and is sent back the parser itself, which can be used to extract the required data). With this approach to storing link indices between databases for supporting queries against a collection of databases, SRS generalizes the traditional querying approach by providing a uniform interface that makes easier the use of exported indices from each data source. Initially, SRS focused on flat-file databases [e.g. SwissProt (http://www.ebi.ac.uk/swissprot/), GenBank, EMBL and PROSITE (http://us.expasy.org/prosite/)] but it now supports the querying of relational data sources and XML data sources. The recently released version 8 of SRS provides a WSDL-based web-services interface.

Introducing web services to represent data sources establishes a standard abstraction over the exact mechanism used to retrieve data, thereby enabling the user to access a variety of data sources in a uniform manner. One major problem that remains is the interpretation of the internal format of the data that are returned. This could be addressed, in part, by making use of the efforts of the Open Bioinformatics Foundation (http://www.open-bio.org/), which produces libraries for Perl, Java, Ruby and Python (called, unsurprisingly, BioPerl, BioJava, BioRuby and BioPython, respectively) that are dedicated to providing common-object formats that map to many popular data sources and applications.

Cross-domain expertise integration

What is, perhaps, one of the most exciting prospects for the use of web services comes from their ability to address the traditional domain fragmentation of the life-sciences field, whereby the databases and tools used within each subdiscipline form isolated islands of expertise. This is illustrated best by the gap that currently exists between bioinformatics and cheminformatics, which contributes to the failure of many promising 'genomics' targets to transit successfully into chemistry and, ultimately, the clinic. Bridging this gap would enable, for example, the connection of gene-expression models with both the chemical properties of the compounds that the samples were subjected to and the gene-sequence data to the level of individual genotypes. There are many technical reasons for the current lack of integration, including data complexity, schema instability, data and tool decentralization, legacy systems and distributed computing resources. In addition, many cultural barriers exist that stem mainly from unwillingness to share data and results.

The benefit of the web-service approach to building cross-domain models is in establishing a common information framework to combine the tasks in different domains.

A typical-use case is a pharmacogenomics pipeline in which single nucleotide polymorphisms and other nucleotide-level changes in one sequence are analysed against high-volume databases of genotypic data, gene–disease association libraries, targeted microarray experiments and clinical information to establish the effectiveness of a particular drug treatment on the analysed genotype. Further to this, the chemical properties of the proteins involved in the relevant pathways are investigated to determine whether some of them might be a better target, thereby shortening the drug-discovery pipeline and reducing the risk of a prospective compound failing in the tox-screening stage.

A service architecture that is properly designed can easily support the required operations with a relatively small set of intercomponent protocols and communication models (sequence data, generic relational data structure and, possibly, profiling representation).

Web-service composition

One of the driving goals of web services is to facilitate the construction of interactive applications that peruse distributed resources to solve ambitious complex tasks. Therefore, the issue of composing web services – enabling different services to communicate with each other – becomes paramount. Web services, themselves, are stateless by design – they do not include a mechanism to capture the continuity between the requests made to a database or the results computed by a tool – and individual tasks in a chained process do not know about each other. However, if there were an analysis process or application that perused these web services and passed the data from one service to another, that application must know about its execution state. To preserve the logic of the entire process, a stateful environment is needed in addition to the stateless model.

Web services are, simply, functional components that receive input data and send output data to each other. Enabling the unrestricted composition of components can be problematic in a generic execution environment. Thus, most such environments enforce restrictions on possible compositions. For example, although two-way exchange of data between components might be possible in theory, it would be used only rarely for scientific analysis. A mechanism enabling strict one-way communication between services is simpler and is sufficient to model the vast majority of scientific analysis scenarios. This model is typically known as a workflow or, more precisely, a dataflow variety of the workflow in which an individual component cannot start executing until all of its input-data dependencies are satisfied.

Data go into one component that performs a relatively coarsely grained analysis task (including domain identification and a homology search) and sends the result to another component, with possible data-format conversion involved, that uses the result and processes it further. Such workflows can be authored using workflow programming languages and submitted for execution to

workflow enactment engines. In this case, individual services are represented as nodes within a workflow, with the data being passed between them at execution time.

In some workflow environments, metadata are used to prevent incompatible services from being joined together. Metadata describe the type and format of the data being passed between services. In contrast to data that are communicated between services at execution time, metadata must be propagated between services at workflow construction time to enable service implementations to commit to consuming and producing the correct data types. Such data types might range from simple standard types covered by SOAP – such as string, numbers and floating-point values – to more-complex domain-specific metadata (descriptors of a relational table with column names and types, sequence collection descriptors listing sequence features, and document collection with keyword and term-annotation data).

Workflow composition languages and standards

Although workflow composition languages and standards for business applications have matured markedly in recent years, this has not been the case for scientific workflow languages. Business workflow systems have been used since the 1970s to track document movement through an organization, assign individual tasks to people and draw dependencies between them. They have since evolved into software tools that handle the automation of e-commerce applications between different enterprises. Different standards for business workflow systems have also matured, for example Business Process Modelling Language (BPML) and, more recently, the Web Service Business Process Execution Language (WSBPEL) standard developed by Microsoft (http://www.microsoft.com/), BEA (http://www.bea.com/framework.jsp?CNT=homepage_main.jsp&FP=/content) and IBM (http://www.ibm.com/us/), and maintained at OASIS. WSBPEL separates abstract processes (business protocol and message behaviour only) from executable ones (including execution details, activities and error handling). BPML is a strict superset of WSBPEL used to describe end-to-end processes; it describes the roles performed by all participants. It separates clearly aspects of control-flow processes from those of data-flow processes, in addition to supporting nested processes.

By contrast, scientific workflow languages and systems are younger and far less tangible. Activities captured in scientific workflows are research tasks rather than business transactions and, thus, are more varied and flexible than their business counterparts. So, when the first generation of web-service standards evolved with BPML and WSBPEL, it was no surprise that the use of these languages fell short of the expectations of research projects. Features such as data management, knowledge capture and service publishing were not represented to a sufficient degree. Meanwhile, commercial and academic scientific-software analysis tools, not necessarily conforming to the standards mentioned, began moving into the provision of workflow capabilities for service composition. Notable examples include InforSense KDE (using Discovery Net technology) and Taverna, which provided scientists with a feasible way of producing meaningful, reusable and understandable representations of their research.

The UK Engineering and Physical Sciences Research Council (EPSRC)-funded Discovery Net (http://www.discovery-on-the.net/) project at Imperial College London (http://www.ic.ac.uk/) provides a service-oriented grid-computing system for knowledge discovery. The system enables end-users to connect to and use various proprietary data-analysis software, data sources and software tools that are available online by third parties through a variety of methods, including web-service interfaces. The Discovery Net system has been used in a variety of projects, including real-time collaborative genome annotation [9], large-scale severe acute respiratory syndrome (SARS) evolution modelling [10], real-time high-throughput cheminformatics and integrative life-science analysis applications [11].

Taverna, which is an outcome of the EPSRC-funded myGrid research project in the UK, provides language and associated software tools that facilitate the use of workflow technology over web services whose inputs and outputs are defined using WSDL. The project uses XScufl, an XML dialect of the Simple Conceptual Unified Flow Language, for communicating results between the services. Taverna is currently used in various life-science projects, including some that are investigating Graves' Disease, Williams-Beuren Syndrome and, more recently, trypanosomiasis in cattle. Taverna also has a running collaboration with BioMOBY in which it uses its semantic descriptors for metadata annotation of services.

InforSense KDE provides a visual environment for constructing and executing distributed scientific workflows based on Data Processing Markup Language (DPML). Its approach is to integrate bioinformatics, cheminformatics, and data-mining and text-mining applications within a single service framework. Each workflow captures and implements the analytical logic of an application that integrates the data and software tools as required by analysts and business users. The workflows can be executed directly from the visual environment or published as web services (in several formats, including WSDL–SOAP) for access by other tools. The system is currently used by several leading pharmaceutical and biotechnology companies, and research centres.

Other workflow systems that should be mentioned include VIBE by Incogen (http://www.incogen.com/vibe), which specializes in sequence informatics, and Turboworx (http://www.turboworx.com/), which provides advanced workflow architecture for basic local alignment search tool (BLAST) executions.

Web-service semantics

In parallel to the evolution of standards that address technical aspects of the web services and their connectivity,

there has been a major evolution in the activities of the semantic web community, with the aim of automating and simplifying complex web-service interactions. The idea underlying such efforts is the production of machine-understandable descriptions of remote web services. The availability of such semantic descriptions would, ideally, enable automatic service discovery and composition to be performed. The approach builds on the use of controlled vocabularies that describe domain-specific concepts of a particular application domain.

Currently, the most widely accepted technology for capturing domain-context knowledge is ontologies. In frameworks of semantic web services, this representation is used to add a formal semantic layer on top of other system elements such as software components and registries, and link them to the terms used in the problem domain. For example, a web-service application might use gene ontology terms to describe tasks performed by individual services and, subsequently, define keywords to be used in indexing the resulting service.

Two major projects are currently tackling this challenge. The first is OWL-S, a joint effort by BBN Technologies (http://www.bbn.com/), Carnegie Mellon University (http://www.cmu.edu/), Nokia (http://www.nokia.com/), Stanford University (http://www.stanford.edu/), SRI International (http://www.sri.com/) and Yale University (http://www.yale.edu/) that is defining ontology for semantic markup of web services. The second is WSMO by the Digital Enterprise Research Institute, which brought together several academic and industrial partners such as SEKT (http://sekt.ijs.si/), DIP (http://dip.semanticweb.org/), Knowledge Web (http://knowledgeweb.semanticweb.org/), Lion and Ontoweb (http://www.ontoweb.org/).

OWL-S approaches the issue by defining three parts to its ontology: profile, process model and grounding. Profile is used for advertising and discovering services (basically, it is a semantically oriented UDDI) and is meant for service-seeking or matchmaking agents. Process model exposes the internal workings of the service to the desired degree and is used for service composition and execution monitoring. Finally, grounding establishes the communication protocols and message formats that the service employs to implement the features described in the process model.

WSMO takes a different approach in that its focus is on composition and usability aspects. Technical communication details are left aside (which might change in the near future – the WSMO authors initially saw that area being well covered by OWL-S) in favour of user-interaction modules, separation between choreography and orchestration of web services (user–consumer perspective and provider perspective, respectively), mediators that handle concept translation between different services, and goal-driven automated service linkage.

The semantic approach to composing web services might seem to be somewhat superfluous in the current service space, but its importance becomes apparent as soon as one considers large-scale service-based systems that rely on multiple providers and multiple matchmaking agents that search for relevant components to fulfill the user's task. One solution would be to employ a single service middleware such as InforSense KDE or Taverna that could handle these problems efficiently within an organization committed to the software, but this is not possible with a fragmented service landscape that can barely agree on a common service publication standard, let alone a common software platform. Capturing most of the complexity within the services themselves, semantic standards represent the next step in web-service evolution, and their acceptance will be the measure of success of service-oriented architectures as a whole.

Available web services in the life sciences

Currently, a transition is underway that should, hopefully, lead to the availability of all life-science resources as web services. Although this is far from happening, most major institutions are already offering a large portion of their tools and databases in this manner. Here, we present the most important ones (the relevant links are given in Table 1).

European Bioinformatics Institute

EBI has published the complete content of its EMBL nucleotide-sequence database within the XEMBL project. This massive database, produced in collaboration with GenBank and DDBJ, contains (at the time of writing) 47 billion entries. The data are offered in two different formats, Bioinformatic Sequence Markup Language (BSML; http://www.bsml.org/) and Architecture for Genomic Annotation, Visualization and Exchange (AGAVE; http://www.lifecde.com/products/agave/). BSML is a format for representing the biological sequence data, and annotations and features on it, together with meaningful references to external data sources. AGAVE represents full genomes and links the manually annotated feature data with expression and regulatory information. XEMBL is available through a SOAP–WSDL interface.

Another project of interest from EBI is the Open Bibliographic Query Service (Open BQS), which, in itself, is the web-service implementation of the BQS standard for searching and retrieving scientific citations. It runs on the Medline database (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=MedLine).

Finally, Soaplab [12] is a project whose goal is to provide standardized wrappers for Perl and Java programs so that they can be accessed as web services. It is supported by the BioPerl and BioJava projects and, on its own, contains no implementations. However, EBI wrapped up several tools – mostly EMBOSS applications – using Soaplab, as proofs of concept. Today, these services are officially supported and available from the Soaplab site (http://www.ebi.ac.uk/soaplab/).

GenomeNet

GenomeNet offers several databases and computational services, among them KEGG, which is available through a WSDL-based web-service interface. In addition to information about metabolic and regulatory pathways, KEGG stores information about the gene products, chemical compounds and reactions that constitute those pathways. The full functionality of KEGG is accessible through the web-services application program interface (API), including the pathways represented in KGML, the XML language for representing the biological networks. Its associated viewer is also available from GenomeNet.

DNA data bank of Japan

DDBJ has been working in collaboration with EBI and NCBI (http://www.ncbi.nih.gov/) to catalogue nucleotide sequences. It is now offering several SOAP services through individual WSDL files: for example, BLAST, ClustalW, DDBJ, Ensembl, Fasta, PML, RefSeq, SPS, SRS and TxSearch. Although it offers a relatively simple selection of tools, a multitude of databases in the background makes it an attractive offering for composing sequence-based service workflows.

Virginia Bioinformatics Institute

VBI and EBI share the honour of being the first institutions to offer their bioinformatics web services to the public. As part of the PathPort [13] project, several tools have been packaged in WSDL–SOAP interfaces and published, including the DAS server over the RefSeq database of the NCBI, Mummer, FASTA and BLAST over a Condor cluster, Glimmer, ClustalW, pathogen information and microarray-normalization algorithms.

It is worth noting that VBI separates its offering into several categories: freely available services are accessible by anyone at any time; no-cost, license-restricted services typically have some limitations associated with them (mostly because of licensing terms of underlying software or data sources); and pay-restricted services require payment using the VBI authentication, authorization and accounting ticket system. Non-production services, which are not currently supported, are available on a 'use-at-your-own-risk' basis.

Although there might not yet be a centralized registry of web services, individual organizations are working hard at publishing their existing applications in WSDL–SOAP-friendly form. This is helped further by programmatic toolsets such as the Soaplab and OpenBio libraries that enable rapid wrapping of command-line programs and easy parsing of different output formats. It is because of these technologies that it is now feasible to construct distributed bioinformatics pipelines that perform genuinely useful and powerful tasks.

National cancer institute

The cancer Bioinformatics Infrastructure Objects (caBIO) consist of an object model and architecture that links a large number of various data sources within the National Cancer Institute (http://www.nci.nih.gov/). SOAP is supported as one of the interfaces for accessing the caBIO functionality. Other methods include Java and Perl APIs whose object models correspond to the ones used in the SOAP interface.

Concluding remarks

During the coming years, there will be an increase in the use of web-services technology in life-science applications. This increase will be driven by the need for a systematic and standardized approach to tools and data repositories used in research environments. However, the efficient use of web services is still hindered by the lack of high-level tools that support the end-user directly; it is worth remembering the software-engineering maxim that 'imperfect technology in a working market is sustainable, while a perfect technology without a market will vanish' [14]. Although web services are far from being the one-stop solution to the problems of life-sciences integration, they have proven to be good enough, in the sense that their presence has already forced people to rethink the way in which their research informatics are working. Whether this technology will be called web services, grid services or something completely different in ten years is irrelevant. The key paradigms established in recent years will remain the foundation for future efforts.

References

1 Al Sairafi, S. *et al.* (2003) The design of Discovery Net: towards open grid services for knowledge discovery. *International Journal on High Performance Computing Applications on Grid Computing: Infrastructure and Applications* 17, 297–315

2 Oinn, T. *et al.* (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics* 20, 3045–3054

3 Wilkinson, M.D. (2004) BioMOBY – the MOBY-S platform for interoperable data service provision. In *Computational Genomics Theory and Application* (Grant, R.P., ed.), Horizon Bioscience

4 Martin, D. *et al.* (2004) OWL-S semantic mark-up for web services (http://www.w3.org/Submission/OWL-S)

5 Roman *et al.* (2004) Web Service Modeling Ontology (http://www.wsmo.org/TR/d2/v1.2/)

6 Berners-Lee, T. *et al.* (2001) The semantic web. *Sci. Am.* 284, 34–43

7 Stevens, R. *et al.* (2001) A classification of tasks in bioinformatics. *Bioinformatics* 17, 180–188

8 Etzold, T. and Argos, P. (1993) SRS – an indexing and retrieval tool for flat file data libraries. *Comput. Appl. Biosci.* 9, 49–57

9 Rowe, A. (2003) The Discovery Net system for high throughput bioinformatics. *Bioinformatics* 19 (Suppl. 1), i225–i231

10 Curcin, V. *et al.* (2004) SARS analysis on the grid. In *Proceedings of UK e-Science All Hands Meeting 2004*, pp. 114–120, EPSRC

11 Gilardoni, F. (2005) Integrated life sciences – an oxymoron? *Journal of QSAR and Combinatorial Science* 24, 131–137

12 Senger, S. (2003) SOAPLAB – a unified sesame door to analysis tools. In *Proceedings of the UK e-Science All Hands Meeting 2003*, pp. 509–513, EPSRC

13 Eckart, J.D. and Sobral, B. (2003) A life scientists gateway to distributed data management and computing: The PathPort/ToolBus framework. *OMICS* 7, 79–88

14 Szyperski, C. *et al.*, eds (2002) *Component Software Beyond Object-Oriented Programming* (2nd edn), Addison–Wesley and ACM Press