
A Unified View of Protein Sequence and Structure Using the Distributed Annotation System

Andreas Prlic¹, Thomas Down¹, Alex Bateman¹, Tim Hubbard¹, Christine Orengo², Mark Dibley², Robert D. Finn¹

¹ Wellcome Trust Sanger Institute, The Wellcome Trust Genome Campus, Hinxton, Cambs, CB10 1SA, UK

² Department of Biochemistry and Molecular Biology, University College London, London, WC1E 6BT, UK

The eFamily project aims to improve integration between protein sequence and structure data. One of the technologies being used to achieve this is the Distributed Annotation System (DAS). DAS is a specific Web service technology that performs the exchange of biological annotations. DAS is motivated by the idea that annotations should be provided by independent decentralised databases. Here, we describe DAS and compare it to other Web service technologies. We describe a registry server for DAS, and Spice, a new DAS client with support for protein DAS and the registry system. Finally, we demonstrate how Spice can be used to annotate proteins with both sequence and structure data.

Background

Proteins are the cogwheels that act together to ensure that the cellular machinery works. They consist of a “string of beads”, where the beads are amino acids. In the cell, this linear chain folds into a 3-dimensional (3D) structure. The precise 3D structure is determined by the sequence of the amino acids. Once proteins have adopted their fold, they can fulfil their function. Experimentally it is relatively easy to determine a protein’s amino acid sequence, but harder to determine its 3D structure. This is reflected by the respective size of the databases - currently the sequence database contains 1.5 million entries, but only 30,000 entries can be found in the database of 3D structures [1,2].

The **eFamily project** [3] brings together 5 of the world’s leading molecular biology databases: **CATH**, **InterPro**, **MSD**, **Pfam** and **SCOP**, all based in the UK [4-8]. These databases are providing protein sequence or structure information. MSD is the European

database that archives protein structures. Pfam and InterPro classify proteins into families based on sequence, whereas SCOP and CATH classify proteins based on their known structure.

The resources for archiving protein sequence and structure have historically been developed independently, leading to difficulties in navigating between the two. As the number of protein sequences and structures exponentially increase the need to integrate the two types of data becomes more pressing. An area of research the eFamily project has been addressing is providing a **unified view of protein sequence and structure** resources.

One technology used for this is the **Distributed Annotation System (DAS)** [9]. The DAS protocol allows data producers to share their results with the community without requiring aggregation into a central database [1]. DAS is a document orientated web service technology. It conforms to the service-orientated architecture (SOA) concept [Figure 1]. With SOA there are three integral components; a service provider, a server requestor (client) and a service registry. As discussed below, we have addressed technical issues with all three components.

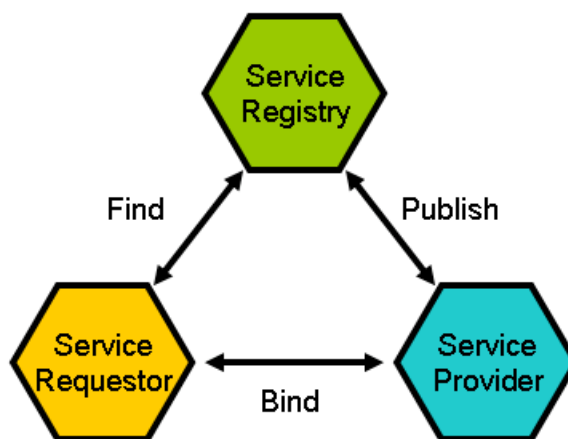


Figure 1 – A service orientated architecture (SOA). There are three components to a service orientated architecture concept: Service provider, service requestor and a service registry. Connecting these components together are three operations (publish, find and bind). Adapted from [12].

The Components of DAS-SOA

The Service Provider

DAS is a technology closely related to Web service technologies [Figures 1 & 2]. The arrangement of a DAS service is virtually identical to other web services [Figure 2]. Within a server, typically Apache Tomcat, there is a servlet container. In the case of Web services that use SOAP [middle, Figure 2], the servlet container has an Axis installation, and this together with the service specific code, produces the web services. However, in the DAS scenario, the Axis installation is replaced by **Dazzle** [10][left, figure 2]. Dazzle is a Java servlet written specifically for the DAS protocol. Dazzle is a modular system which uses small “data-source” plugins to provide access to a range of databases. These plugins provide the means to easily provide a new set of data using DAS.

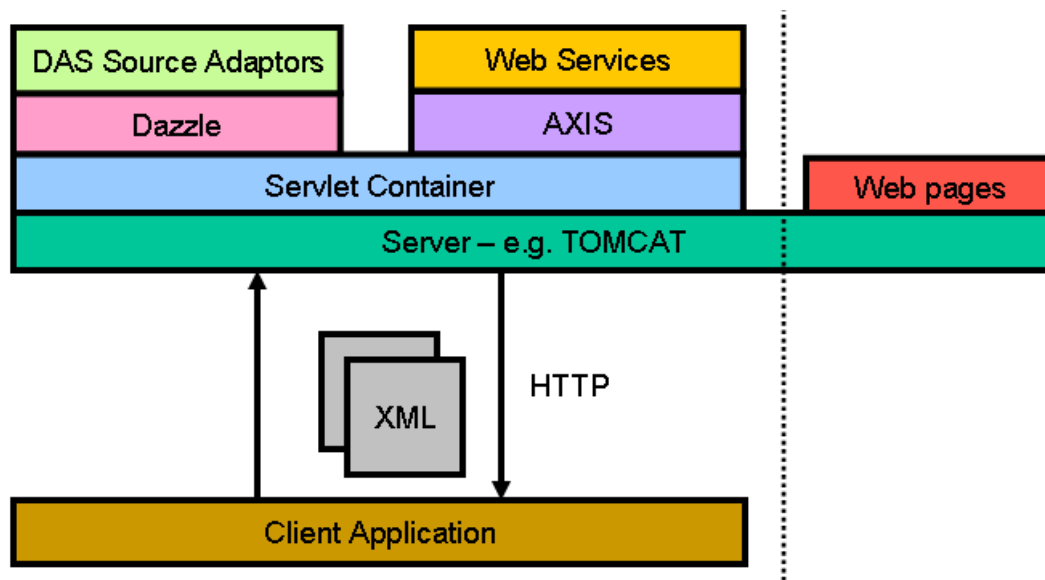


Figure 2 – Comparison of the organisation of DAS and Web Services on the server. See text for details.

In both the DAS and Web Services responses take the form of XML documents transported over http (Figure 2).

Unlike Web Services that can take any form, with DAS, there are two main types of servers, **reference and annotation servers**. Reference servers provide the biological objects to be annotated, like protein sequences or structures, while annotation servers provide the actual features of annotations.

The Service Registry

A typical problem of decentralised Web service based technologies is the discovery of service providers. We have developed a **registration service** that lists available DAS services. DAS clients can connect to this registration service and retrieve a list of available services. The registration service itself

validates DAS servers and regularly tests their availability. If a server becomes unavailable it is labeled as such. Servers that have been down for a longer period of time will be removed from the public listing. The server administrators can receive a notification email. The registration server is implemented as a standard web-service, see:

http://das.sanger.ac.uk/registry/services/das:das_directory?wsdl

The Service Requestor/Client

In the DAS protocol, the clients are the workhorses in this system. The client obtains a reference object and subsequently retrieves annotations for this objects from (multiple) DAS servers. A DAS client requests data by requesting data from a URL, which follows a particular syntax. E.g. the

following URL requests all annotations of the sequence with accession code P00280.

<http://www.ebi.ac.uk/das-srv/uniprot/das/aristotle/features?segment=P00280>

DAS servers process the request and responds with an XML document as defined by the DAS specification (for a detailed specification see [9] or

<http://www.biodas.org>). It is then the role of the client to combine the reference and annotations object to a human readable format, typically a graphical representation. In the DAS system, the CPU load is predominately on the client side, rather than the server side.

A working example: SPICE - DAS client

Having described the component of the DAS system, it is appropriate to describe how a client works in reality. To do this, we will use the **SPICE DAS client** that has been produced as part of this project as an example [13,14]. The biological data that the eFamily project deals with can be communicated in different coordinate systems (these can be considered as different languages). These can be genomic DNA sequence, protein sequence or protein structure. The information in which coordinate system data is provided is stored on the DAS registration server. DAS clients can use this information

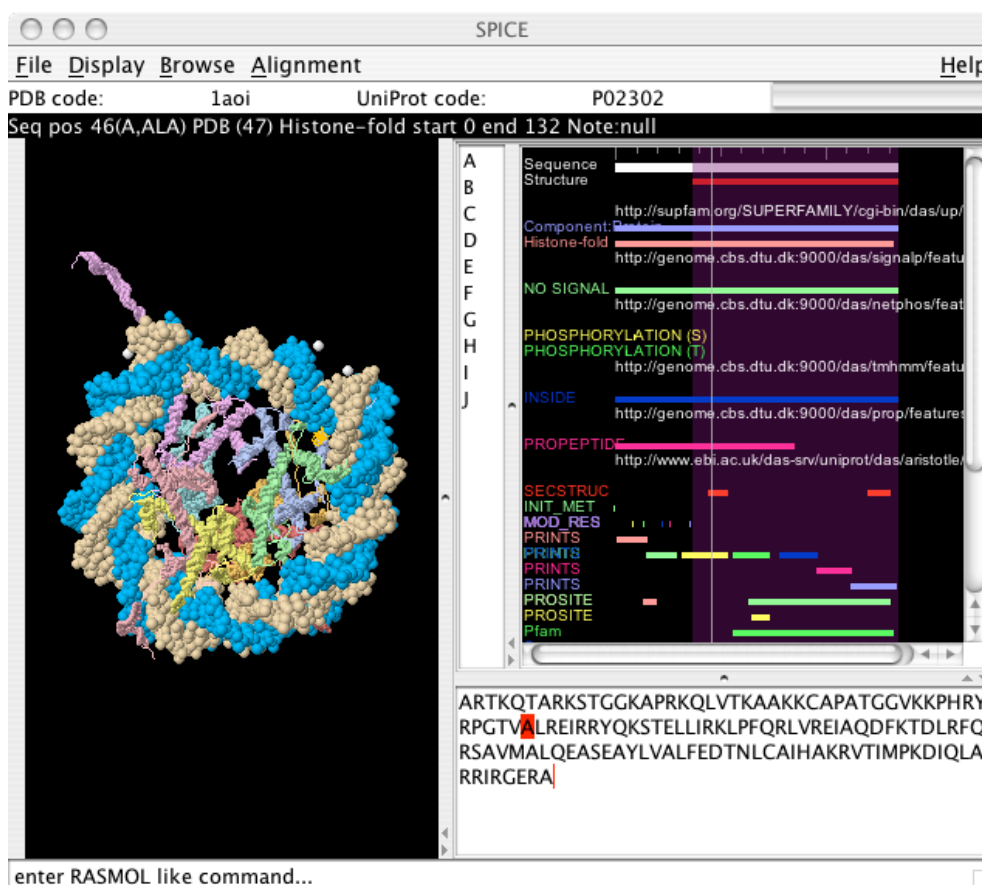


Figure 3: The SPICE client allows users to browse through annotations of protein sequences and structures. It retrieves data from different sites across the Internet via the DAS protocol.

to request data only from servers providing data in the suitable coordinate systems or can identify the different coordinate systems being used and in-turn request a **mapping service between the coordinate systems**.

Each of the member databases of the eFamily project makes a contribution to the data available via DAS (the publish operation in Figure 1). The MSD database provides a residue by residue mapping between the sequence (UniProt) and structure database. Protein annotations are provided by the CATH, SCOP, Pfam and InterPro databases.

To visualize the data the SPICE client can represent both protein sequence and structure annotations [Figure 3]. SPICE can be started via the Internet using **Java Web Start** technology. Once it is running it connects to the DAS registration server using Web Services to retrieve all available DAS service providers (the find operation in Figure 1).

When a user requests a protein structure, by a PDB identifier, the client retrieves the 3D coordinates of protein. The SPICE client can either connect to a local replica of PDB or make a DAS request to a protein structure server to obtain the 3D coordinates (the bind operation in figure 1). For DAS to supply 3D coordinates we use the DAS extension that we have written.

As we want to display protein annotations based on both protein structure and sequence, the client also makes a DAS request to get the corresponding sequence objects

and mappings, while the 3D coordinates are being retrieved. The mapping server provides the alignment between the protein sequence and structure objects. At the time of writing, the UniProt to PDB alignment used in SPICE is based on the MSD mapping. If other alignments are made available the user can choose between alternative servers and compare the provided alignments.

Finally the **annotation features** are retrieved from the annotation servers. These can be provided in either protein sequence position or PDB residue number coordinates. The client now has all of the information and performs the reciprocal mapping between the different coordinate systems such that the all features can be displayed on the protein structure [left, Figure 3] and on the protein sequence [right, Figure 3].

The 3D structure of a protein may have been determined several times and a structure can consist of several sequences. To deal with this many to many relationship, a window is available that allows users to choose which of the alignments to display.

Future developments include provision of more data using the DAS protocol, especially multiple sequence and alignment data as provided by the project member databases. SPICE will be extended to support browsing not only through individual protein sequences and structures, but also through the network of related protein sequences and structures.

SPICE is available under the LGPL. For the 3D visualization SPICE builds on the Jmol

library [15]. Both Jmol and SPICE are available under the LGPL. Different open source viewers are available for visualization of protein sequence or structure data, e.g. **Jalview** [18] or **PFAAT**[19], to mention two implemented in Java. These could be easily extended to support DAS by incorporating the DAS-client side libraries.

Integration with the Scientific Community

DAS is an established protocol in the genome bioinformatics community; in particular it has been used extensively by **Ensembl** to exchange data as part of their genome visualisations [11]. To make the DAS protocol applicable for protein structures we developed two extensions to the DAS specification that allow alignments and 3D structure information to be transmitted. See <http://www.eFamily.org.uk> for protein DAS schema extensions [3], which is becoming referred to as protein-DAS. Nevertheless, due to the generic nature of these extensions, they have been used by other areas of the community. For example DAS can now be used to exchange genomic DNA alignments. Evidence of the versatility of DAS as a data grid technology means that it is not restricted to the member groups of the eFamily project. Consequently other academic groups have started to provide their data using protein-DAS and at the time of submission there are 16 protein sequence and structure DAS servers from 5 different European countries.

So, why is DAS widely used? It is **simple to use and simple to set up**. This is partly due to the fact that both DAS-servers and client solutions are supported with implementations in multiple programming languages:

- Java - Dazzle
<http://www.biojava.org/dazzle/>
- Perl - Proserver
<http://www.sanger.ac.uk/Software/analysis/proserver/>
- Perl -LDAS
<http://www.biodas.org/servers/LDAS.html>

When new annotations become available, these can be published as a new DAS service by readily adapting simple DAS adaptor plugins to suit the new data. DAS-client side libraries are also available via the Biojava and Bioperl projects.

Furthermore, biological annotations are usually made freely available to the scientific community, so there are no data security issues.

Collaborative work within the DAS community and the myGrid project [16] is currently underway that aims to allow DAS server/client solutions to become compatible with Taverna workflows [17]. Two ways of interaction are currently being investigated: A) if a DAS-client requests data from a DAS-server this triggers a workflow in the background. Once the workflow has finished, the DAS-servers caches the compiled results for

future requests. B) A workflow that annotates a set of protein sequences or protein structures can automatically provide results as a DAS server. Taverna workflows and DAS-clients visualisation techniques are complementary and this interaction will extend the possibilities for both sides, in particular, taking advantage of other Web services provided by the eFamily project [7] •

Availability

SPICE if available under the LGPL.

<http://www.efamily.org.uk/software/dasclients/spice/>

References

1. H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne: *The Protein Data Bank*. Nucleic Acids Research, 28 pp. 235-242 (2000)
2. Bairoch A, Apweiler R, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N, Yeh LS (2005) *The Universal Protein Resource (UniProt)* Nucleic Acids Res. 33: D154-159.
3. Robert D. Finn, Andreas Prlic, Ujjwal Das, Phil McNeil, Nicola Mulder, Sameer Velankar, Antonina Andreeva, Dave Howorth, Mark Dibley, Tim Hubbard, Rolf Apweiler, Kim Henrick, Alexey Murzin, Christine Orengo, Alex Bateman. *eFamily: Bridging Sequence and Structure*. Proceedings, UK e-Science, All Hands Meeting 2004
4. Pearl F, Todd A, Sillitoe I, Dibley M, Redfern O, Lewis T, Bennett C, Marsden R, Grant A, Lee D, Akpor A, Maibaum M, Harrison A, Dallman T, Reeves G, Diboun I, Addou S, Lise S, Johnston C, Sillero A, Thornton J, Orengo C. (2005) *The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis*. Nucleic Acids Research. Vol. 33 Database Issue D247-D251
5. Mulder NJ, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, Bradley P, Bork P, Bucher P, Cerutti L, Copley R, Courcelle E, Das U, Durbin R, Fleischmann W, Gough J, Haft D, Harte N, Hulo N, Kahn D, Kanapin A, Krestyaninova M, Lonsdale D, Lopez R, Letunic I, Madera M, Marsden J, McDowall J, Mitchell A, Nikolskaya AN, Orchard S, Pagni M, Ponting CP, Quevillon E, Selengut J, Sigrist CJ, Silventoinen V, Studholme DJ, Vaughan R, Wu CH. (2005) *InterPro, progress and status in 2005*. Nucleic Acids Res. 33, Database Issue: D201-5.
6. Velankar, S., McNeil, P., Mittard-Runte, V., Suarez, A., Barrell, D., Apweiler, R. and Henrick, K. (2005) *E-MSD: an integrated data resource for bioinformatics*. Nucleic Acids Res. 33 (Database Issue): D262-D265
7. Alex Bateman, Lachlan Coin, Richard Durbin, Robert D. Finn, Volker Hollich, Sam Griffiths-Jones, Ajay Khanna, Mhairi Marshall, Simon Moxon, Erik L. L. Sonnhammer, David J. Studholme, Corin Yeats and Sean R. Eddy *The Pfam Protein Families Database* Nucleic Acids Research (2004) Database Issue 32: D138-D141
8. Andreeva A., Howorth D., Brenner S.E., Hubbard T.J.P., Chothia C., Murzin A.G. (2004). *SCOP database in 2004: refinements integrate structure and sequence family data*. Nucl. Acid Res. 32: D226-D229
9. Dowell RD, Jokerst RM, Day A, Eddy SR, Stein L. *The distributed annotation system*. BMC Bioinformatics. 2001;2:7. 2001.
10. <http://www.biojava.org/dazzle/>
11. <http://www.ensembl.org>
12. Steve Graham, et.al. Sams. ISBN: 0672326418; *Building Web Services with Java: Making Sense of XML, SOAP, WSDL, and UDDI*, 2nd Edition
13. <http://www.efamily.org.uk/software/dasclients/spice/>
14. Andreas Prlic, Thomas Down, Tim Hubbard *Adding some SPICE to DAS*. Bioinformatics, in press.
15. <http://jmol.sourceforge.net/>
16. <http://www.mygrid.org.uk/>
17. <http://taverna.sourceforge.net/>
18. <http://www.ebi.ac.uk/~michele/jalview/>
19. <http://pfaat.sourceforge.net/>