



Talisman—rapid application development for the grid

Thomas M. Oinn

EMBL European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Essex, CB10 1SD, UK

Received on January 6, 2003; accepted on February 20, 2003

ABSTRACT

In order to make use of the emerging grid and network services offered by various institutes and mandated by many current research projects, some kind of user accessible client is required. In contrast with attempts to build generic workbenches, Talisman is designed to allow a bioinformatics expert to rapidly build custom applications, immediately visible using standard web technology, for users who wish to concentrate on the biology of their problem rather than the informatics aspects. As a component of the My-Grid project, it is intended to allow access to arbitrary resources, including but not limited to relational, object and flat file data sources, analysis programs and grid based storage, tracking and distributed annotation systems.

Contact: tmo@ebi.ac.uk

INTRODUCTION

Current workbench packages deliver a high level of functionality accompanied by a correspondingly high level of complexity. Systems such as SRS and the Ensembl web interface are both highly capable and rather intimidating to even a relatively experienced user. Many potential users of these systems are unable to access the functionality they require because of this complexity barrier; this is a problem that is not likely to be overcome, these systems are inherently complex and there is little that can be done to reduce the complexity without a corresponding reduction in expressive power and functionality.

There is a significant class of users who, whilst requiring a complex set of analysis and data access tools, do not have the experience to develop their own or to use these complex workbench systems without significant extra training. This especially applies to lab biologists who wish to analyze their data; often this data analysis is conceptually complex, involving many disparate resources. In theory, this is exactly the kind of application that grid technology is suited for, however, there is currently no simple way of accessing these grid resources. Effectively, we need an analog to the web browser for accessing web pages; such a client must be simple to use and simulta-

neously powerful enough to be useful, with reference to the paragraph above, this would seem to be an intractable problem.

Talisman does not attempt to provide a workbench system for general use. Instead it provides a highly expressive framework that may be used by expert bioinformaticians to produce ‘canned web based applications’. These resultant applications, although they may access the full complexity and capability of the grid and other similarly complex systems, are themselves simple to use and fairly narrow in their functionality, and thus well suited for use by scientists wishing to focus more on the biology than the informatics. The support infrastructure within Talisman allows this process to be sufficiently agile to make this approach practical; without such a system the required time and effort to generate new tools often prevents it ever happening.

TALISMAN APPLICATIONS

A Talisman application consists of one or more XML definition files. These files define the behavior of the application in terms of data encapsulation (nouns) and actions (verbs). The definition syntax is intentionally minimalist, relying significantly on Talisman’s expert systems for page layout and information presentation. Most production applications are written in under four pages of XML definition; typically this compares to some thousand lines of java source code to perform the same function. When accessed by a user via their web browser, these definition files are compiled into a server side data structure, which is then presented as a set of dynamically generated web pages allowing the user to interact with this data structure, the interactions and the data shown are both completely defined by the original XML definition file.

Talisman provides various core data types and actions, such as text fields, images, regular expression based text operations and similar generic functionality. The real power of the system comes in the extensibility mechanisms; novel systems require a small amount of bridging code to be written to allow access from within Talisman applications, however, once this code has been

written, the novel system can suddenly cooperate with everything else that exists within Talisman. For example, we have written the bridging code to allow Talisman to make use of Martin Senger's SOAP web service wrapper around the EMBOSS tool set, the code to do this was written in two days.

Talisman's runtime environment performs various functions. The most obvious is to run the applications that you give it, however it also includes additional functionality that all applications may make use of. The most significant at the present time is that of provenance tracking.

PROVENANCE TRACKING

As a user accesses any Talisman application, a provenance track is generated which records both the results of the applications, and metadata concerning versions, parameters, arbitrary user annotations etc. The intention is to allow someone to go back to an *in silico* experiment they may have run in the past, and see answers to the following questions:

- What did I do?
The provenance log stores enough information for a user, whether human or software based, to reconstruct the process that was run.
- When did I do it?
Every node in the provenance track contains a time stamp, so it is trivial to see exactly when and in what order processes occurred.
- How was the process run?
Metadata about the applications accessed by the Talisman application such as SQL strings, BLAST parameters etc are held in the provenance track, along with any additional information that Talisman can glean as it runs. Ideally, metadata such as database versions would be stored, of course, this is dependant on the system accessed being able to provide this information, and Talisman can only collect information that is available.
- Who ran the process?
This question starts to become meaningful when these provenance data are deployed to any kind of distributed lab book or storage system; although Talisman itself has no native concept of a user having to sign on (as many applications would not require this), it is possible to collect user authentication data through a suitable plugin.
- Why was a particular choice made?
Talisman applications are by definition interactive. Often, the user will make a decision that involves years

of training and experience, but that to an application is completely opaque. For example, the choice of one particular endonuclease out of a set of several appears to the application as a simple text picking operation devoid of any semantic payload. To cover these situations, any point of interaction within a Talisman application may be flagged by the author as annotatable; when a user encounters such a point the application will offer an opportunity to annotate the event with arbitrary text, which is then stored in the provenance log.

As the provenance track by itself is just a piece of data, Talisman also interacts with the MyGrid information repository module to store this information, and may interact with the MyGrid ontology service to classify the experiment represented by this provenance track within a descriptive logic for indexing.

TALISMAN PLUGINS

In addition to the core Talisman actions and data types, the following plugin packages are either completed or under development. We anticipate that by ISMB2003, all the following will be available as production quality code:

- RDBMS access
This set of action plugins allows Talisman to read and write data held in relation database management systems. Talisman can handle collection of login data, connection pooling, auditing and other basic relational functionality.
- Web, .net and Grid services
General web services may be called from Talisman using this package, allowing use of services distributed over the net. While any given service obviously requires specific configuration to access, this package simplifies the process of doing so. Example services that may be targeted include BLAST and BQS (Bibliographic Query Service).
- Ensembl integration
This package contains data types and actions that allow full access to the Ensembl data set, including sophisticated set based searches and data retrieval.
Bioinformatics service access
This package allows access to the full set of EMBOSS functionality via the SOAPLAB package written by Martin Senger. This becomes especially powerful when used in conjunction with the Ensembl access package, allowing for sophisticated bioinformatics workflows to be built with remarkably little effort.

- Interaction with the MyGrid project components

Talisman is currently developed under the auspices of the MyGrid project, and as such can act as a web interface to some of the higher level MyGrid functionality. In particular, by default Talisman will interact with the MyGrid information repository to store provenance data and intermediate results in a form suitable for subsequent use by distributed lab book style tools. The last are currently under development, we anticipate that they will be running and stable by the time you read this.

TALISMAN SERVER PROPERTIES

All Talisman applications are hosted within a Talisman server. The user accesses the generated applications using a standard web browser; there is no client side installation at all. The servers exhibit various desirable properties:

- Hot deploy and update

Applications running under a Talisman server are simple to update, there is no need for server restarts and no interruption to active users of the application when it is updated, the users will simply see the new version of the application the next time they start it. This is particularly important for production environments where it may not be practical to ask a large number of people to stop using an application whilst it is updated.

- Persistent server state

Dependant upon the configuration of the servlet engine that Talisman resides within, user sessions can be persisted over server restarts, this also potentially allows for fault tolerant clustering if a third party distributed session manager is installed. It is possible to configure a Talisman installation to be extremely resilient and fault tolerant.

- Platform independent

Any server that can run a java servlet container can run Talisman; it has been tested on win32, i86 linux, MacOS X and Solaris.

- Layers on top of existing open standards

Talisman itself is an open system licensed under the GNU General Public License (GPL), and makes use of technology that is similarly licensed. It is possible to build an entirely functional fault tolerant Talisman cluster using commodity hardware and open source operating systems and server software.

AVAILABILITY AND COLLABORATION

Sourceforge.net, from which release builds may be downloaded, hosts talisman development. A running server is available at the European Bioinformatics Institute for those who are curious to see an installation in action (obviously, not all functionality is available), and all code is held in CVS, which may be browsed or accessed anonymously from the sourceforge site. The relevant URLs are as follows:

- Talisman test installation at the EBI (<http://www.ebi.ac.uk/collab/mygrid/service1/talisman>)
- Talisman project page on sourceforge.net (<https://sourceforge.net/projects/talisman/>)

We welcome any offers of collaboration, especially if you have a system that you think could sensibly integrate with Talisman. This applies in particular to any groups with novel software applications that could potentially have increased functionality if integrated with other similar applications.

ACKNOWLEDGEMENTS

This work is supported by the UK e-Science programme EPSRC GR/R67743, & DARPA DAML subcontract PY-1149, Stanford University.

The author would like to acknowledge the *my*Grid team: Matthew Addis, Nedim Alpdemir, Rich Cawley, Vijay Dialani, David De Roure, Alvaro Fernandes, Justin Ferris, Rob Gaizauskas, Carole Goble, Kevin Glover, Chris Greenhalgh, Mark Greenwood, Karon Mee, Peter Li, Xiaojian Liu, Phillip Lord, Darren Marvin, Simon Miles, Luc Moreau, Norman Paton, Steve Pettifer, Milena Radenkovic, Angus Roberts, Alan Robinson, Tom Rodden, Martin Senger, Nick Sharman, Robert Stevens, Paul Watson & Chris Wroe.

REFERENCES

- Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
- Etzold,T., Ulyanov,A. and Argos,P. (1996) SRS: information retrieval system for molecular biology data banks. *Methods Enzymol.*, **266**, 114–128.
- Rice,P., Longden,I. and Bleasby,A. (2000) EMBL: The European Molecular Biology Open Software Suite. *Trends Genet.*, **16**, 276–277.
- Stevens,R.D., Robinson,A.J. and Goble,C.A. (2003) MyGrid: personalised bioinformatics on the information grid. In *Proceedings of the Eleventh International Conference on Intelligent Systems for Molecular Biology*.