The Data Playground: An Intuitive Workflow Specification Environment

Andrew Gibson[1], Matthew Gamble[1], Katy Wolstencroft[1], Tom Oinn[2], Carole Goble[1]
1. *School of Computer Science, The University of Manchester, Manchester, UK*
2. *European Bioinformatics Institute, Hinxton, UK*
andrew.p.gibson@manchester.ac.uk

Abstract

Workflows systems are steadily finding their way into the work practices of scientists. This is particularly true in the in silico science of bioinformatics, where biological data can be processed by Web Services. In this paper we investigate the potential of evolving the users' interaction with workflow environments so that it more closely relates to the mode in which their day to day work is carried out. We present the Data Playground, an environment designed to encourage the uptake of workflow systems in bioinformatics through more intuitive interaction by focusing the user on their data rather than on the processes. We implement a prototype plug-in for the Taverna workflow environment and show how this can promote the creation of workflow fragments by automatically converting the users' interactions with data and Web Services into a more conventional workflow specification.

1. Introduction

Bioinformatics is a discipline in which e-Science can, and is, making a big difference [1-3]. The field is still relatively new, but can be characterized as being carried out in an almost completely *in silico* environment, and by the need to access and analyze potentially very large datasets that are stored in many different repositories [3]. Bioinformatics emerged as a response to the need to detect scientifically important signatures buried in the ever increasing amounts of biological data. As such, the early tools of bioinformaticians were bespoke and produced using high level programming languages such as Perl. The favored way of making these tools accessible were as scripts running on servers that were accessed with forms embedded in Web pages. As bioinformatics has become more important to the life sciences as a whole, the way in which bioinformaticians interact with biological data and analytical tools has moved beyond the cottage industry scale.

In this paper we focus on the approaches taken by two related projects, myGrid and its workflow environment, Taverna [1,4], and myTea[1]. We identify a strong potential for extending the Taverna workflow environment so that it is possible for a user to conduct two modes of bioinformatics work:

1. A *routine process-based mode* in which a task is well defined by the desired outcome (data-signature) and the steps necessary to get to that outcome, e.g. an annotation pipeline
2. An *investigative data-oriented mode* in which the overall outcome is unknown at the outset, and the connections between the steps are governed by the data outcome of each step.

The first mode has clearly identified use cases [5, 6] and is well supported by workflow environments such as Taverna. The second mode, characterized by the myTea project, is less well-served by mainstream e-Science platforms, though examples like [7, 8] exist. We show that the concept of a data-oriented mode is not only an intuitive mode of interaction for bioinformaticians, but that it can also be seen as the pre-processing that leads to workflows.

To validate the concept, in the second half of this paper we describe the specification and development of a Taverna plug-in, the Data Playground. This plug-in, which applies principles required to perform the second mode of work, is designed to encourage the user to experiment by running data through available and appropriate Web Services. It also features the ability to "record" a set of tasks performed on some data, which can be subsequently automatically converted into a conventional process model workflow.

2. Background

Many workflow environments have been developed including Taverna [1], DiscoveryNet [9], Triana [10]

[1] http://www.mytea.org.uk

0-7695-3064-8/07 $25.00 © 2007 IEEE
DOI 10.1109/e-Science.2007.19

and Kepler [11]. In this study we use Taverna for two reasons; it has an extensible plug-in architecture and it already supports the BioMoby framework, which we use as the means to access the data services that constitute the steps in our *in silico* investigation. BioMoby has a repository of typed Web Services which have semantic descriptions that can be exploited as being unambiguous with respect to each other [12].

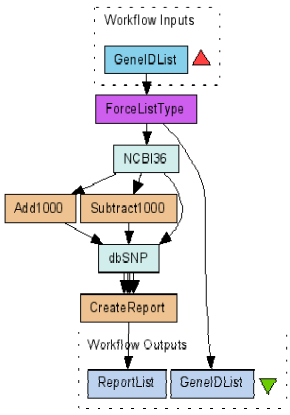

Figure 1: A sample Taverna workflow that represents processes, their inputs and outputs and connections to other processes.

Like other workflow systems, Taverna has aimed to take advantage of the programmatic access granted by Web Services to:

- Provide automation support for bioinformaticians who repeatedly run a series of processes manually, often through many different web-based interfaces.
- Provide integrated access to data stored in many different repositories and in many different formats and allowing high throughput data processing.

Taverna provides an interaction environment and automates the transfer and conversion of data between processes through the specification of workflows, as seen in Figure 1. The immediate advantage of this approach is an effort-saving one, as the bioinformatician is released from the need to repeat mundane tasks. The standard HTML form based tools of bioinformatics are not easily strung together, and require the user to copy and paste relevant data between form elements. Workflows comprised of Web Services remove the need for this type of action.

Effort saving is not the only advantage that users of Taverna have discovered. A number of additional advantages have become apparent such as: workflows can be scheduled to run at regular intervals; the provenance of data passing through a workflow is assured and can be retrieved at a later date; the integrity of data passing through a workflow is safeguarded from human error. Moreover, the workflow is a clear, unambiguous and explicit record of the process – the experiment provenance.

2.1. From the routine to the exploratory

The myTea project explored the support for day-to-day tasks of bioinformaticians through the provision of an electronic lab-book. A significant finding was that the aspects of bioinformatics that require the most support are those that elicit the digital traces of *in silico* experimentation. Digital traces in bioinformatics exist in files on the hard drive as data, notes or results. Many of these are lost as the bioinformatician visits multiple online Web pages or use local programs for which they may not be able to capture or record all of the appropriate information necessary to repeat the experiment.

Crucially, the level of investment needed to perform *in silico* experimentation is so low that the amount of time a bioinformatician would have to invest in carefully recording all of the details would practically take longer than the experimentation itself. Also, there is often little motivation to record negative results, as it is much easier to move on to the next task for which there may be a positive outcome.

Consequently, bioinformaticians will often know that a re-run of an *in silico* experiment can be performed if the need arises. Re-running an analysis provides no guarantee that the result will be the same as the previous one however, as many datasets are constantly updated. Often, if a result is to be recorded, it will be in a condensed semantic form; "the experiment X gave me the result Y". The desirable situation would be that such statements could always be supported with the original data, or at least the specification of how to re-acquire the result from some initial dataset – again, the experiment provenance.

Clearly these two modes of work, routine and exploratory, are complementary and share requirements that should be supported in a common environment. The problem originally addressed by myGrid and Taverna was of bioinformaticians analyzing data from disparate resources with multiple steps that needed to be repeated at regular intervals. The steps would originally have been most accessible through forms in Web pages [6]. The design for the solution to the problem was to create an environment in which the processes that comprised the overall task could be bolted together to accomplish it automatically and programmatically.

The myTea project identified that during exploratory analyses for interesting data-signatures, a bioinformatician may go through a fast cycle of hypothesis generation, result evaluation and method selection (Figure 2). Once a data signature has been

identified, a process of *post hoc* rationalization is needed to validate the results. Subsequently the process can be formalized, which is a prerequisite for workflow description.

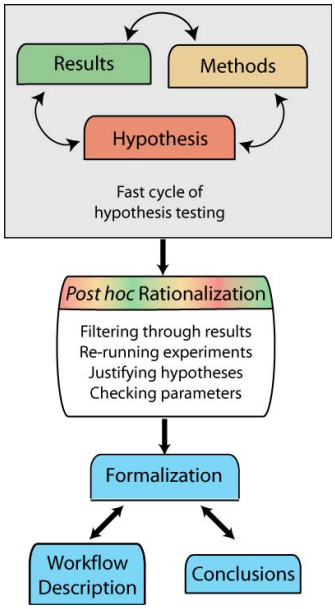

Figure 2: The *ad hoc* cycle of experimentation that precedes identification of interesting data-signatures and potential workflow specification.

2.2. Work practice evolution

The creation of workflow environments has initiated an evolution of work practice for bioinformaticians who are aware of their existence. The original Taverna use case targeted a cohort that had achieved a very clear understanding of their own task and goals, but perhaps through a lack of any viable alternative. These scientists would have exhaustively optimized their work practice of retrieving data and processing it through web pages. However, a significant amount of bioinformatics is performed in modes that do not have such well defined goals. The introduction of such users to a workflow environment like Taverna can be counter-intuitive.

As workflow environments and Web Service components have been developed and evolved, so have the available possibilities for the work practices of bioinformaticians. The approach of the myTea project was to provide a data management environment, providing the bioinformatician with direct access to Web Services to retrieve and process data. This was designed to be a much more intuitive introduction to an advanced interface environment that would reproduce the effects of interacting with a Web page, but with the benefits of recording the details of the action that are not possible through a Web page.

The two approaches are at extremes in terms of what the user is trying to achieve; Taverna as it stands supporting a well defined task and outcome, myTea supporting *ad hoc* experimentation with little or no predetermined plan or intended result. The evolution of intermediates of these extremes should support a spectrum of bioinformatics practices with the potentials being demonstrated with new technologies. As experience with these technologies increases, the demand for support for different work practices along the spectrum will increase.

The Data Playground lies within this spectrum as a workflow development support environment that provides a more intuitive method of specifying workflows by allowing the user to experiment with the effect of using Web Services on data.

3. Requirements for the Data Playground

The Data Playground concept draws on existing advantages inherent in the Taverna environment, and some of the more intuitive aspects for users outlined by the myTea project. The key requirements for a Data Playground environment are outlined below.

3.1. Service discovery

The Data Playground should support the discovery and navigation of Web Services based on the data that the user has focused on. It is prudent to remember the complexity of modeling bioinformatics analysis. In the absence of a well characterized set of tasks from which conventional workflow design can be initiated, there are a large number of options available to the investigator that depend on both the data at hand and the analyses that can be performed on that data. Also, the huge quantities of biological data available to a bioinformatician come in many different types and formats.

In a space of so many possibilities, modeling or anticipating how a bioinformatician might behave could get unmanageable very quickly. Instead the system needs to guide the user to what moves it would be possible and sensible for the user to make. Different types of data are compatible with different types of analyses. Data contained in results from analyses have a similar diversity. To navigate these requires the user to a) be able to discover available Web Services, b) ascertain if the purpose of the process is desirable and c) be informed as to what formats the inputs and outputs of the Service are.

3.2. A data-oriented interface

The Data Playground should support the more intuitive mode of interaction by promoting a data-centric view.

The concept presented here relies on access to Web Services through the workflow environment, but in a way that is driven by the user interacting with some familiar biological data. Consequently, the user should experience a more intuitive ability to invoke particular Web Services to transform or analyze the data.

Taverna specializes in support for modeling the processes that make up a workflow. Workflow development presents a rather static view of processes by attempting to produce a pre-determined outcome. For a bioinformatician in an exploratory work mode, it is possible to take data and attempt to identify interesting signatures within that data by using Web based bioinformatics resources. The transformations between data types and formats are achieved manually or also through predominantly Web-based tools.

It is important that this data-centric methodology does not require some prior knowledge of the desired outcome, and the user has the freedom to explore the possibilities of moving the data through the appropriate processes. This is in stark contrast with the traditional static view of processes in a workflow environment.

3.3. Workflow specification support

The Data Playground should include features that improve the transition from *ad hoc* data-oriented experimentation to process oriented workflow description.

It is not the aim for the user to require *a priori* knowledge of what components the workflow might end up having or even what the main outputs of the workflow might be. However, subsequent to the user having made informed, *ad hoc* decisions about running data through a set of processes in the Data Playground, they may realize that a certain combination of processes was useful, and that it is likely that this combination of processes will be useful again. As such they will want to convert a set of processes from the data-oriented playground view to the standard process oriented Taverna view. This should be as automated as possible so as to not put up any unnecessary barriers of comprehension to the user.

4. Data Playground prototype plug-in

To validate the requirements of the Data Playground, we have implemented a prototype plug-in for Taverna, as seen in Figure 3. The plug-in supports the services provided by the BioMoby project. The Taverna workbench has already been extended with the BioMoby framework, enabling the use of their semantic discovery components. BioMoby Web Services have typed semantic descriptions that can be exploited as being unambiguous with respect to each other [12], making it necessary for the user to specify a BioMoby type for any data that they bring into the system. Web Services "in the wild" typically have not been designed to work together. The adoption of BioMoby meant we could avoid the complication of format transformation services (known as "shims") between services.

Taverna lacks a dedicated data manager, meaning that the user has to have some external interaction with a data repository or locally stored files that contain data. In a full implementation of a Data Playground this could have implications for the integrity and the provenance of data.

Finally, we captured the users' actions as an active process rather than a passive one. Ideally, the Data Playground would passively monitor and record all user activity such that a workflow could be defined based on a successful outcome. The specification of a method for picking out the appropriate subset of processes from a very noisy history of all processes is non-trivial, as seen in systems like [13,14,15], and we were not sure it was even required.

Despite these limitations, this left plenty of scope to demonstrate the plug-in and validate the concept.

4.1. Implementation

The prototype contains several packages. The Playground Model package is implemented through a series of extensions to the Java Universal Graph and Network (JUNG) Framework[2]. The User Interface package contains both a number of extensions to the JUNG visualization framework for the Playground Panel implementation and also the implementations of the additional user interface panels. Communication between the user interface and the Taverna Workbench is conducted through the set of defined Software Programmable Interfaces (SPIs).

The visualization and interface to the model is implemented as an extension to the visualization

[2] http://jung.sourceforge.net/

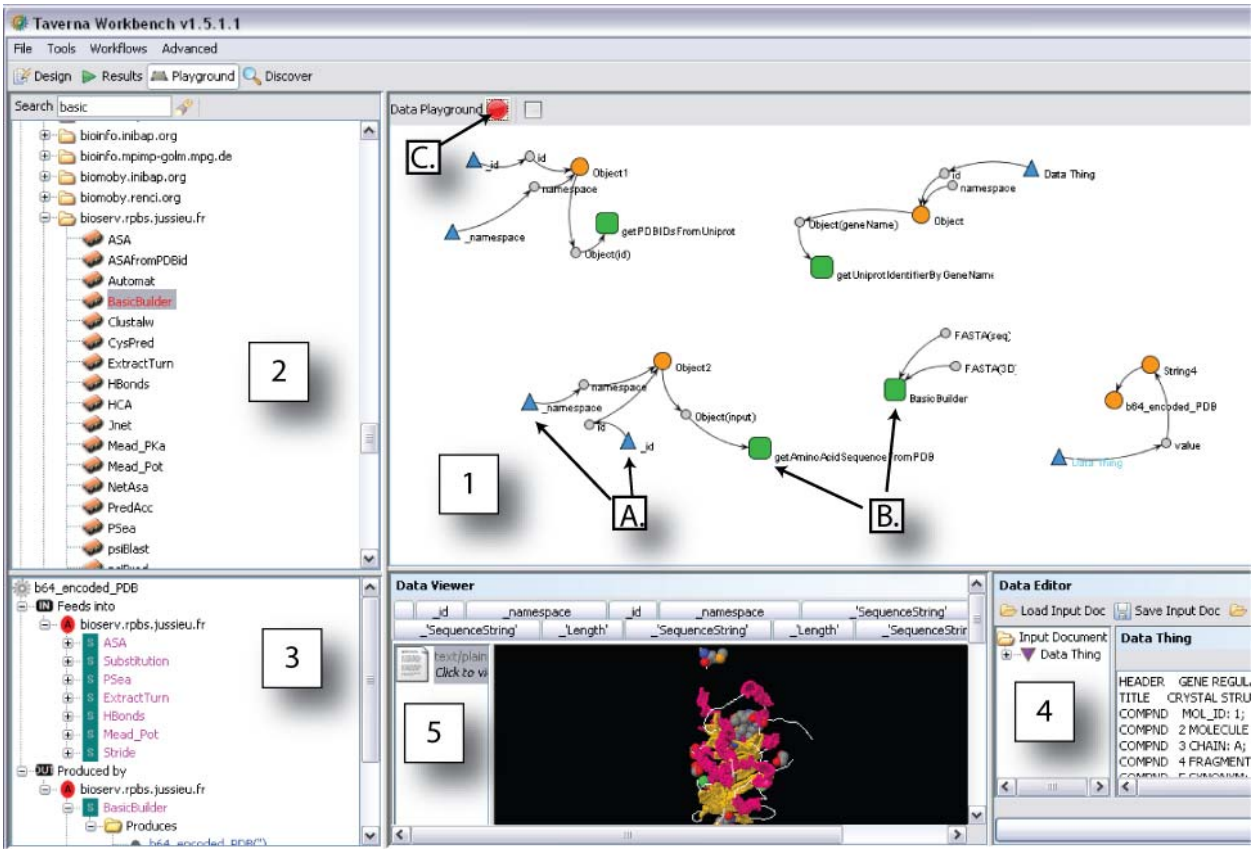

Figure 3: The prototype data playground plug-in in Taverna. 1) The Playground Panel with A) Data Objects that feed into BioMoby objects, B) BioMoby Services which can be executed through a right-click option and C) the 'record' button for recording user activity. 2) The Taverna tree of Web Services which can be dragged into the Playground Panel and connected to Data Objects. 3) Semantic Discovery panel that shows compatible services based on input or output. 4) The Data Editor panel for inputting Data. 5) The Data Viewer panel for viewing data that features the Taverna data rendering capabilities.

framework also provided by JUNG. It separates the notion of model and view by providing both an extendable graph framework and a separate visualization framework. The JUNG visualization framework was considered to be the most suitable of a few considered because of low runtime memory requirements and the ease of integrating it into the JUNG graph framework.

The prototype of the Data Playground can be installed using the Taverna plug-in manager from:

http://www.mygrid.org.uk/dataplayground

4.3. User interface

The Data Playground user interface appears in a tab within the Taverna view as shown in Figure 3. The main component is the "Playground Panel" that acts as a canvas for creating objects and links between graphical Playground objects. These include objects that represent data, parameters, processes and results. Data can be introduced by creating new data objects in the Playground Panel and adding in bioinformatics data using the data editor panel. Similarly data can be inspected using the data viewer. Processors comprise a third area of the screen and can be introduced by dragging them onto the Playground Panel. Data is connected to process-inputs by simply dragging a connection between them.

4.4. Data-oriented viewpoint

The prototype follows the requirements by putting emphasis onto data as something that can be manipulated and processed. The Data Objects are easy to create, are always visible and can be inspected by the user. This creates a simple but crucial link between Web Services and data that is missing in the traditional view of Taverna where it would not be so simple to try and just run a Web Service to see what would happen,

meaning that the bioinformatician would be forced back to more conventional Web page interfaces.

4.5. Semantic discovery

Another key requirement for the Data Playground is that it is 'smart' in terms of the types of data. This means restricting the options for processes based on what analyses it would be sensible to perform given the type of data. In the prototype, this ability manifests itself through the "get processor details" function of BioMoby. Using this, the user can see which services are compatible based on the type of input and output of the selected service. In the case of the prototype this is only possible because of the typing system in BioMoby, but proves an extremely valuable operation to perform and even aids to highlight Web Services that the user may not have known were available for their analysis. Again, this only serves to encourage the user to explore the analysis options associated with their data.

4.6. Record actions and convert to workflow

The Data Playground includes a 'Record' function that, once activated, monitors and records the users' activity. Once the user has settled on a set of processes, a new playground can be initiated and the record function can be activated by the user. The Data Playground then begins to store the processes that the user selects as they run their analyses. When the recording is stopped, the plug-in automatically converts the moves that the user made into a conventional workflow that can be saved and used like any other workflow.

5. Playground experiences

Four experienced Taverna users participated in a series of design and evaluation sessions giving direct feedback. A wider panel of a further 7 users and developers participated in an evaluation workshop, open discussion and completed questionnaires.

Although only an experimental prototype, the approach was considered sufficiently promising by our evaluation panel of Taverna users and developers to warrant further investigation.

Clearly, there is plenty of room for improvement. The lessons learned will feed into the next iteration of the Data Playground to better fit the specification. Also, as users are recruited, its functionality can be extended to better represent its desired mode of use.

Passive vs Active recording. The prototype features a record function that has an active mode of operation. The conversion of a set of actions to a workflow proved both viable and intuitive. This was sufficient for a proof of concept design, but some serious consideration is needed for how to make this into a passive system that records all actions, yet can differentiate meaningful actions. Work by [15] is promising in this regard. Passive recording would mean that the user would not have to repeat a sequence of analyses that proved to be useful after the fact, and instead would be able to select from an overview of their past actions. Additional requested functionality such as pause and undo suggest that the active record and the passive mode are not mutually exclusive.

Service compatibility. The Web Services were restricted to BioMoby Services because these are compatible and their signatures are semantically typed. Metadata will be needed for a much larger list of services. There is still a way to go before enough are sufficiently annotated so that a user can rely on a system to intelligently guide them [17]. myGrid has a full-time curator annotating the most popular Web Services with an ontology [16]; however, service providers need to contribute to the effort to improve scalability. An alternative is to enforce a stricter typing regime on the participating services, an approach adopted within systems such as DiscoveryNet [9], or within well-defined and managed scientific enterprises such as caBIG [2]. Techniques for identifying components needed to map between incompatible data, such as in [18] will also need to be incorporated.

Service metadata through play. Many BioMoby service providers had poorly or incorrectly annotated their services. A side effect of the playground is that the attempted use of a service with data, the user may discover a more appropriate annotation for the service inputs. By providing the user with the mechanism to annotate the service with the available Moby object types we facilitated the gathering of further metadata about the service.

Data management. Data in the prototype is transitory with respect to the processes, which means that the user has to retrieve or access some data that is remote to the Playground and type it before analysis can begin.

Multiple instances. Users asked for the ability to run multiple instances of the playground. This may further encourage the type of experimentation that bioinformaticians are accustomed to by being able to compare the outcomes of multiple canvases and

allow several lines of investigation to be followed at the same time. This may alleviate the limitations of the real-estate on a canvas.

Scalability. The approach is not intended be used to generate very large workflows in one go. Instead it is expected to be used to develop sub-workflows that will be stitched together as components in a larger workflow. Neither is it intended to handle large-scale data; the data are representative examples of that to be managed on an "industrial scale" by the workflow.

6. Related work

There is a plethora of research and experience in the practice of observing a user's interactions as they undertake a task, recording this as a script or macro and playing it back. We were inspired by "Programming by Example" [19,21], automatic macro systems [22,23,26] and programming systems for non-programmers [20]. Principles such as ensuring everything relevant to the current task is visible on the screen, and maximizing point and click over textual input were useful to the design of the prototype.

Workflow by example is a less well-developed area. The WbE system [24] does observe the user, but builds workflows across a database rather than across multiple services and does not support direct data manipulation on the desktop. The Composition Analysis Tool [25] and [27] focus on intelligent user guidance during workflow composition, but still within a conventional process-oriented rather than data-oriented workflow assembly approach, with a view to workflow consistency. Galaxy [8] is an example of a workbench that offers facilities to interact directly with data through a conventional web browser, forms-based interface but has yet to generate workflows, although BioManager [7] does. Frameworks such as Chickenfoot [13] can be used to construct workflows over web-based forms. The Data Playground is not a browser or forms based system. We chose instead to use a direct manipulation paradigm. Direct manipulation interfaces are a well-developed area including innovative commercial products such as Surface [28], and Looking Glass [29] which we would love to try out.

7. Final thoughts

Bridging the gap between technology and the work practices of expert groups such as those in the life sciences remains a significant challenge. e-Science must rise to this challenge by creating new concentrations of expertise from where it is much less effort for both technologists and scientists to reach common understanding of requirements and solutions. The Data Playground tries to harness the strong technological advantages of a workflow system and combine this with in-depth understanding of work practices.

Bioinformaticians are becoming aware of the advantages of expert systems emerging from e-Science, even if these are not always those they expected. The expertise and experience of the early-uptake users is sufficient for them to see how they might specify their own evolutions in existing technologies that might help them with tasks they perform now, or even enable them to perform work that they had previously not considered possible.

The Data Playground is one of a number of evolutions for workflow interaction. The forthcoming release of myExperiment [30] is another example. It provides scientists with an unprecedented community aspect to workflows, promoting their sharing and re-use and encouraging collaboration. This represents a trend where scientists are no longer expected to start the process of workflows specification with a blank canvas. Instead, they can learn from the experience and expertise of others, reusing and adapting whole workflows or fragments of workflows [31]. The Data Playground provides a complementary environment where workflow fragments are auto-generated from the scientists' more natural mode of data exploration. The future seems to rely on the interoperability of such workflow-related environments to provide their users with the flexibility to carry out their normal work practices whilst advancing the benefits of workflows to researchers in the life sciences.

Acknowledgements

We acknowledge the myGrid and myTea teams, and all our users. This work is funded through UK grants: EPSRC EP/D044324/1, EP/C536444/1, GR/R67743/01 and EP/E502997/1.

References

[1] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn, "Taverna: a tool for building and running workflows of services," Nucleic Acids Res, vol. 34, pp. W729-32, 2006.

[2] K. K. Kakazu, L. W. Cheung, and W. Lynne, "The Cancer Biomedical Informatics Grid (caBIG): pioneering an expansive network of information and tools for collaborative cancer research," Hawaii Med J, vol. 63, pp. 273-5, 2004.

[3] L. D. Stein, "Integrating biological databases," Nat Rev Genet, vol. 4, pp. 337-45, 2003.

[4] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, and P. Li, "Taverna: a tool for the composition and

enactment of bioinformatics workflows," Bioinformatics, vol. 20, pp. 3045-3054, 2004.
[5] P. Li, K. Hayward, C. Jennings, K. Owen, T. Oinn, R. Stevens, S. Pearce, and A. Wipat, "Association of variations on I kappa B-epsilon with Graves' disease using classical and myGrid methodologies," Proc 3rd UK e-Science All Hands Meeting, Nottingham UK, 2004.
[6] R. Stevens, H. J. Tipney, C. Wroe, T. Oinn, M. Senger, P. Lord, C. A. Goble, A. Brass, and M. Tassabehji, "Exploring Williams-Beuren Syndrome Using myGrid," Proc 12th Intl Conf on Intelligent Systems in Molecular Biology, Glasgow, UK, Bioinformatics, 20:i303–i310, 2004.
[7] BioManager Australian National Genomic Information Service (ANGIS) http://www.angis.org.au, 2007
[8] B Giardine, et al Galaxy: A platform for interactive large-scale genome analysis Genome Res. 15: pp:1451-1455, 2005
[9] A. Rowe, D. Kalaitzopoulos, M. Osmond, M. Ghanem, and Y. Guo, "The discovery net system for high throughput bioinformatics," Bioinformatics, vol. 19, pp. i225-231, 2003.
[10] D. Churches, G. Gombas, A. Harrison, J. Maassen, C. Robinson, M. Shields, I. Taylor, and I. Wang, "Programming scientific and distributed workflow with Triana services," Concurrency and Computation: Practice & Experience,, 2006.
[11] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock, "Kepler: an extensible system for design and execution of scientific workflows," 2004.
[12] M. Wilkinson, H. Schoof, R. Ernst, and D. Haase, "BioMOBY successfully integrates distributed heterogeneous bioinformatics Web Services. The PlaNet exemplar case," Plant Physiol, vol. 138, pp. 5-17, 2005.
[13] Michael Bolin et al. Automation and customization of rendered web pages. In UIST' 05, pp163–172. ACM Press, 2005.
[14] W van der Aalst, T Weijters, and L Maruster. Workflow mining: Discovering process models from event logs. In IEEE Trans on Knowledge and Data Engineering, Vol 9, pp 1128–1142, September 2004.
[15] TA. Lau and DS. Weld. Programming by demonstration: An inductive learning formulation. In Proc IUI '99, pp145–152. ACM Press, 1999.
[16] K. Wolstencroft, P. Alper, C. Wroe, P. Lord, R. Stevens, and C. Goble, "The myGrid Ontology: Bioinformatics Service Discovery," International Journal of Bioinformatics Research and Applications (IJBRA), vol. 3, pp. Article in press, 2007.
[17] P Lord, S Bechhofer, M Wilkinson, G Schiltz, D Gessler, CA Goble, L Stein, D Hull. Applying semantic web services to bioinformatics: Experiences gained, lessons learnt. Proc ISWC2004, Springer LNCS 3298: 350-364
[18] K Belhajjame, SM Embury, NW Paton, R Stevens, and CA Goble Automatic Annotation of Web Services Based on Workflow Definitions. in ISWC2006, LNCS 4273: 116-129 (2006)
[19] H Lieberman Your Wish is My Command: Programming By Example, Morgan Kaufman 2001
[20] D. C. Smith, A. Cypher, and J. Spohrer, "KidSim: programming agents without a programming language " Commun. ACM, vol. 37, pp. 54-67 1994.
[21] A Cypher Watch What I Do MIT Press, MA, USA, 1993, ISBN 0-262-03213-9
[22] DR Olsen JR Dance Macros by example in a graphical UIMS IEEE Computer Graphics and Applications, 8(1) pp: 68-78, 1988
[23] A. Safonov, J. Konstan, and J. Carlis. Towards Web Macros: a Model and a Prototype System for Automating Common Tasks on the Web. Proc 5th Conference on Human Factors & the Web, 1999.
[24] A Tomasic, RM McGuire, B Myers Workflow by Example: Automating Database Interactions via Induction, CMU-ISRI-06-103 Technical report, September 2006
[25] J Kim, M Spraragen, and Y Gil. An intelligent assistant for interactive workflow composition. In Proceedings of the International Conference on Intelligent User Interfaces (IUI-2004). ACM Press, 2004.
[26] A Sugiura Y Koseki Internet Scrapbook: Automating Web Browsing Tasks by Demonstration Proc UIST '98. San Francisco, CA 9-18 1998 ACM 0-58113-034-1/98/11
[27] Chen L., Shadbolt N.R., Goble C.A, A Semantic Web Based Approach to Knowledge Management for Grid, Applications, IEEE Trans in Knowledge and Data Engineering, 19(2) pp. 283-296 February 2007
[28] Microsoft Surface http://www/microsoft.com/surface
[29] Sun Microsystems, Project Looking Glass, http://www.sun.com/software/looking_glass 2007
[30] CA Goble, D De Roure myExperiment: social networking for workflow-using e-scientists, Proc WORKS '07: 2nd workshop on Workflows in support of large-scale science, Monterey, USA, pp:1-2, 2007
[31] A Goderis, P Li and C Goble. Workflow discovery: the problem, a case study from escience and a graph-based solution. Proc. of the 4th IEEE Int. Conference on Web Services (ICWS 2006), Chicago, USA, 12-22 Sept 2006