Participatory Design of Pipeline tools and Web services in bioinformatics

Catherine Letondal, Olivier Amanatian

Participatory Design of Pipeline tools and Web services in bioinformatics.

Catherine Letondal Olivier Amanatian
Institut Pasteur
Paris, France
letondal@pasteur.fr oliviera@pasteur.fr

January 6, 2004

1 Introduction

In this position paper, we describe a project for developing tools to support biologists using bioinformatics programs and data within a distributed architecture. This architecture relies on the use of Web services as a layer for simple and interactive pipelines, and as a means to help in services discovery. The design of the software is rooted in a study of biologists' work practices with bioinformatics tools that we observed during interviews. A video brainstorming workshop provided us with information about critical features of the future system. After a brief introduction about existing e-science projects in bioinformatics, we describe the outcome of the interviews and workshop and the challenges to face in the technical architecture we envision.

2 Distributed Computing in Bioinformatics

The development of computing tools for biology and genomics has increased at a fast pace to deal with huge genomic data and the need of algorithms to discover their meaning. A major part of these tools are available on the Web, either to provide access to a database, or as a convenient user interface of a program. This situation leads to several problems for the biologist, in particular to combine the use of several tools: each requires a different data format leading to many copy and formatting operations.

2.1 GRID Computing and Web services

Electronic workbenches, such as the NCSA Biology Workbench [Sub98] or W2H [SFG+98] aim to provide an environment to help the biologist maintain his or her data and to find and combine tools adapted to each type of data. These tools however do not provide minimal workflow or pipeline construction.

Important and dynamic projects illustrate the efforts currently made to provide tools for distributed computing and integration: biopipe [HRC+03] for building flexible pipelines, bioperl [SBB+02] or biopython [CC00] for developing components, including remote execution and parsing ones. [Ste03] describes technological attempts to integrate various data sources and services on a common architecture. Likewise, the BioMoby [WL02] and myGrid [SRG03] projects aim to develop Web services and ontologies.

2.2 Mobyle project

Our project, called Mobyle, still in the design phase, relies both on previous software developments, such as Pise, a Web interface generator for programs on Unix [Let00] and on available components, such as bioperl or biopython. In particular, we intend to integrate the BioMoby project and to start with G-Pipe, a tool to build workflows, developed by external contributors on top of Pise.

One of the main objectives of this project is however, with a participary design approach [GK91], to address the following issues:

- Do these technologies solve the problems that actually arise in the combination of tools and in the setting of software protocols for the biologists?
- Are actual use problems addressed or even identified at all in these technological approaches?

In the following, we describe the studies, interviews and workshop, that we have conducted at the Pasteur Institute in order to get a deeper view on these issues.

3 Interviews

About 20 interviews have been organized during the last two monthes. We first contacted biologists by telephone to explain our project. We choosed biologists having used the Pasteur Institute Web server recently, as well as biologists having a significant activity in bioinformatics. Interviews were informal: we just asked the biologists to play before us a scenario of a recent bioinformatic analysis. All interviews have been videotaped. During these interviews, we made several observations:

- *Need for a stable and predictible set of known tools.*

  Most of the time, biologists prefer to use a technique or a language that they already know, rather than a language that is more appropriate for the task in hand. Since work at the bench can sometimes result in unpredictable outcomes, biologists generally tend to prefer tools that they control. For instance, we observed biologists who:

  - use outdated DOS programs, whereas they have Windows installed, or use outdated bioinformatics package on Unix instead of their improved version on the Web, not because they don't like the Web but rather because they know the Unix version,
  - stay within a given Web server providing an apparently exhaustive set of tools, eventhough better tools exist elsewhere.

  Moreover, because Web tools tend to evolve unexpectedly, some biologists we have met prefer to install software on their local computer. This way, they better control the changes.

- *Dealing with equivalent objects.*

  It seems that biologists quite often maintain or have to use several versions of "equivalent" objects, which might be difficult to deal with. For instance:

  - same data files in different formats: they have to be kept, because tools for data (e.g sequence) format conversion *change* these data (see 4),
  - software versions: we discussed with a biologist who keeps several versions of a phylogeny inference program, just in case one of the features of an older version has disappeared in the more recent one,
  - data: one of the biologists we met had to deal with two versions of an annotated genome (mosquito); the issue for him was not to lost too much time in re-analysing the same data,
  - file and printout: the same object, either biological data, analysis result or software documentation, is often kept in two (or more) forms, e.g on the disk and on the paper.

- *Interactive nature of some tasks.*

  Analysis tasks supported by computer tools are not always automatic. Indeed, the biologist has either to check the accuracy or significance of a result, such as a score in a database homology search, and to decide to carry on an analysis according to complex criteria that are not possible to automate, or to extract subsets of data before proceeding. Moreover, the biologist has often to edit intermediate results, that are produced in a format that is not recognized by other tools, and automating such edition would require a little programming.

- *Anticipating.*

  Constructing a pipeline is similar to a programming activity, and programming is by nature anticipating. Pipeline tools, of course, aim to help users. In spite of this fact, users are often sceptical regarding sophisticated but difficult to learn systems: they have to be able to anticipate the benefits that they will draw from them. Users also need to anticipate their own needs, in order to perform an action that will help them in the future.

  With respect to anticipation, we observed several type of behaviours about:

  - anticipating the utility of a software for use or reuse,
  - bookmarking: biologist we met often decide not to bookmark an online tool - they seemed to be confident about the retrieval of the site,
  - saving: biologists often save temporary results, or alternative data formats,
  - customization: some bioinformatics tools, such as SeWer [Bas01], enable the users to easily customize Web forms; however, very few biologists actually use them.

- *Constructing and organizing results: annotation, classification, naming, assemblies.*

  The vast majority of biologists we observed maintain a quite organized record of their analyses. Their files are carefully named after their content, the directories are often organized in accordance with species, genes and experiments names. The data, parameters and results that matter for the research activity are often assembled not only in the notebook, but also in Word files. Biologists also annotate printouts and keep them in classified folders.

- *Data flows.*

  General flows of data belong to diverse categories: input to output of a program, piping of an output to the input of another program, reformatting, transforming, filtering, extracting [SGBB01]. The most often used flow of data we observed is however by copy-and-paste, and this is not really supported by any smart tool [PK97].

- *Search and Retrieval.*

  Usually biologist do bibliographic work on a regular basis, but this is not the case with bioinformatics tools. One would expect that biologists, belonging

to a fast evolving field, would try to discover the latest new tools in order to improve their work. Only one or two of the people we discussed with did a technological survey, about 3 or 4 times a year.

However, this does not mean that search and discovery tools, such as Google or Web service directories are not needed: they are. But they tend to be rather used to find an object (data or program) that has already been used.

4 Video Brainstorming Workshop

There were four groups of six biologists, and four designers. Among biologists, half were trained bioinformaticians, having a significant knowledge in computing. The workshop was co-organized with the leader of a project to build an augmented laboratory notebook [MPL+02], which has several issues in common with our project: organisation of work, building of protocols, just to name a few.

Participants first freely put ideas on large paper sheets. Then, 8 rough paper prototypes of selected ideas were videotaped. The main topics developed by biologists were:

- *reusing executed commands as a script*: all groups played with the idea of reusing executed commands either as an history, a macro or a script, although in a different way. The history or macro was either a simple text file, or a colored list, where the user can remove, edit or re-order commands, and attribute colors to them, according to their importance. One group envisioned a kind of temporal strip where icons representing data and programs appeared as the actions actually occurred.

- *using and defining a pipeline*: ready-to-use analysis pipelines are a popular idea among biologists, similar to their bench protocols. Two groups, rather composed of bioinformaticians, were interested in the definition of pipelines by the end-user. One of them built a prototype of a visual bench where programs and input data types were connected together in a graphical editor. After execution, clicking on a program's outgoing link enabled the user to access intermediate results. Like several existing tools, such as the Biology Workbench [Sub98], Pise [Let00], or BioMoby [WL02] this pipeline editor is data type oriented: the user does only see compatible programs or data types.

  So, unlike the biologists interested in editable histories of commands, this group prefer to *anticipate* the definition of a sequence of command by using a sophisticated editor.

- *dealing with unwanted data transformation*:

  [SGBB01] explains that bioinformatics tools can be classified either as: filters, transformers, collections or forks. Unfortunately, transformers or filters often produce results that are not convenient for the biologists. For instance, they truncate data names - and names are very important for organisation and scientific matters - or they remove part of the transformed data without any notice. Two prototypes addressed this problem. One of them showed how to deal with name truncation by means of menus to select among alternative names produced during the various steps. In this prototype, tags on a phylogenetic tree replacing the original names of species could be changed at will by the user. Another prototype dealt with the unwanted removing of a part of a database entry.

- *dealing with desirable data transformation*:

  As exemplified by a prototype showing the extraction of a subpart of a sequence alignment, and as we observed during interviews, interactive manipulations, on the other hand, are also necessary and cannot be automated.

- *need of a global synthetic view on analyses*: the option to visualize several related analyses' results in a coherent way has been proposed. This meets our observation of assemblies of analysis data during the interviews. In a similar category, one group proposed the idea of visualizing the state of parallel analyses in a global window, in order to compare results: for instance the user would clone a subwindow to start a fork to explore an equivalent analysis.

5 Conclusion: Challenges

The observations made during interviews and the features highlighted during this first workshop raise several issues. Simple problems related to the data and programs' results seem to be more important for biologists not having much training in computing than sophisticated tools. Although apparently humble, these problems are however not so simple on the computing side. Another important aspect is the need for the user to be able to interact with the system at any step, in order to be able to visualize, select and edit intermediate results. Annotation is critical: at which level should it happen, could it be supported by workflow tools without loss of information? How to support different flow levels: Web, computer, printout, paper or electronic notebook [MPL+02]? Discovery and search of new tools: this is maybe not that much needed, although a support has to be provided to restrict discovery according to available data. An important challenge is that very few standard although low usable browsers are omnipresent. Biologists often complain about: weak interactivity, less controls when errors, data security, speed, unexpected changes, limited size for data and analysis results by email. But tools *are* available on the Web.

References

[Bas01] Malay Kumar Basu. Sewer: a customizable and integrated dynamic html interface to bioinformatics services. *Bioinformatics*, 17(6):577–578, June 2001.

[CC00] Brad Chapman and Jeff Chang. Biopython: Python tools for computation biology. *ACM-SIGBIO Newsletter*, August 2000.

[GK91] Joan Greenbaum and Morten Kyng. *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, New Jersey Lawrence Erlbaum Associates, 1991.

[HRC+03] S. Hoon, KK. Ratnapu, JM Chia, B. Kumarasamy, X. Juguang, M. Clamp, A. Stabenau, S. Potter, L. Clarke, and E. Stupka. Biopipe: a flexible framework for protocol-based bioinformatics analysis. *Genome Research*, 1363103, 2003.

[JBD03] W. Jones, Harry Bruce, and Susan Dumais. How do people get back to information on the web? how can they do it better? In *Proceedings of INTERACT 2003, September 1-5, 2003 - Zrich, Switzerland*, pages 793–796, September 2003.

[Let99] Catherine Letondal. Résultats de l'enquête sur l'utilisation de l'informatique à l'institut pasteur. Technical report, Institut Pasteur, Paris., apr 1999.

[Let00] Catherine Letondal. A web interface generator for molecular biology programs in Unix. *Bioinformatics*, 17(1):73–82, 2000.

[MPL+02] Wendy E. Mackay, Guillaume Pothier, Catherine Letondal, Kaare Begh, and Hans Erik Srensen. The missing link: augmenting biology laboratory notebooks. In *Proceedings of the 15th annual ACM symposium on User interface software and technology, Paris France*, pages 41 – 50. ACM Press, November 2002.

[PK97] Milind S. Pandit and Sameer Kalbag. The selection recognition agent: instant access to relevant information and operations. In *Proceedings of the 2nd international conference on Intelligent user interfaces, Orlando, US*, pages 47 – 52, 1997.

[SBB+02] Jason E. Stajich, David Block, Kris Boulez, Steven E. Brenner, Stephen A. Chervitz, Chris Dagdigian, Georg Fuellen, James G.R. Gilbert, Ian Korf, Hilmar Lapp, Heikki Lehvslaiho, Chad Matsalla, Chris J. Mungall, Brian I. Osborne, Matthew R. Pocock, Peter Schattner, Martin Senger, Lincoln D. Stein, Elia Stupka, Mark D. Wilkinson, and Ewan Birney. The bioperl toolkit: Perl modules for the life sciences. *Genome Research*, 12(10):1611–1618, 2002.

[SFG+98] M Senger, T Flores, K Glatting, P Ernst, A Hotz-Wagenblatt, and S Suhai. W2h: Www interface to the gcg sequence analysis package. *Bioinformatics*, 14:452–457, 1998.

[SGBB01] Robert Stevens, Carole Goble, Patricia Baker, and Andy Brass. A classification of tasks in bioinformatics. *Bioinformatics*, 17(2):180–188, February 2001.

[SRG03] Robert D. Stevens, Alan J. Robinson, and Carole A. Goble. mygrid: personalised bioinformatics on the information grid. *Bioinformatics*, 19(Suppl. 1):i302–i304, July 2003.

[Ste03] L. D. Stein. Integrating biological databases. *Nature Reviews, Genetics*, 4(5):337–345, May 2003.

[Sub98] S Subramaniam. The biology workbench: A seamless database and analysis environment for the biologist. *"Bioinformatics", Proteins, 32:, 1998.*, 32(1):1–2, July 1998.

[WL02] MD Wilkinson and M. Links. Biomoby: an open-source biological web services proposal. *Briefings in Bioinformatics*, 3(4):331–341, December 2002.