ENHANCING DATA SHARING IN COLLABORATIVE RESEARCH PROJECTS WITH DASH

THOMAS E. FERRIN, CONRAD C. HUANG, DANIEL M. GREENBLATT,
DOUG STRYKE, KATHLEEN M. GIACOMINI, AND JOHN H. MORRIS

Departments of Pharmaceutical Chemistry and Biopharmaceutical Sciences University of California San Francisco, 600 16th Avenue,
San Francisco, CA 94143, USA

We describe a software framework, called DASH, that enables the facile access, maintenance, curation and sharing of computational biology data among collaborating research scientists. The DASH event-based framework enables members of team-based research projects to describe the multistep computational processing pipelines frequently required to generate data for sharing, monitors multiple distributed data stores for changes, and will then automatically invoke the appropriate processing pipeline(s). These pipelines can be used to communicate the results of data analyses to collaborators using mechanisms such as Web Services. We describe the overall design of the DASH system and the application of a simple DASH prototype to a collaborative pharmacogenomics research project involving several dozen researchers located at several different sites—the UCSF Pharmacogenetics of Membrane Transporters project.

1. Introduction

In a collaborative, team-based research project, each group must be able to share results with others and access data generated by others. Traditionally, this has been done by exchanging data via electronic mail or file transfer. While a more facile approach is to use a shared database, this often incurs the challenge of properly maintaining data integrity in the presence of updates by multiple researchers. For example, in a database used for computational biology, adding and altering data may require the invocation of additional computational protocols that automatically update all related—and especially derived—information. When performed manually, this tedious curation process can act as a deterrent, limiting either the number of participants or the growth of the database, or even preventing a collaborative project from being realized in the first place. While several technological solutions to support collaboration, such as workflow and data integration, already exist, no single solution addresses the needs of collaborative computational biology without a significant expenditure of money or personnel. The goal of the project we describe here is to create a system that enables the facile sharing of data, and is specifically targeted at small- to medium-sized collaborative computational biology projects. We believe this represents a very important class of collaborative science projects, as was discussed at the “Models of Team Science” session [1] at last year’s BECON 2003 Symposium on Catalyzing Team Science [2]. “Team science” and the formation of integrated research networks are also common themes within the NIH Roadmap Initiatives [3, 4], the success of which depends crucially upon the sharing of research data. This fundamental need to efficiently and effectively share research data provides the motivation for the DASH project.

2. User Requirements

The core of collaborative science is the exchange of data among cooperating research groups. However, before any sharing can occur, participants must first agree on *what* data will be exchanged (e.g., experimental data, analysis results) and *how* exchanges will happen (e.g., data format, transfer media). When only a few data sets must be shared, data preparation may be done manually. However, as the number and types of data sets increase, the shortcomings of manual preparation, such as human error during processing and dependence on vigilant monitoring of available data, can become serious hindrances to the collaboration. Timely sharing of data becomes even more difficult if complex and time-consuming manual manipulation of data is needed. For small- to medium-sized academic laboratories, data preparation for collaborations can prove to be quite challenging due to limited funding and staffing. Tools are needed to help streamline data preparation and sharing by addressing the requirements listed in Table 1.

Table 1. User requirements for managing data exchange with collaborators.

R1	*Document data exchange protocol*. Having protocol documentation will ease transitions such as staff or student turnover, which is a particularly difficult problem for a small group where the person leaving may also be the single person who handles all data preparation.
R2	*Facilitate changes in protocol*. In a lengthy collaboration, new types of data may be acquired or analyzed over the course of the collaboration. As the data domain evolves, so must the data exchange protocols.
R3	*Support multiple collaborations*. Multi-group collaborations are becoming more common as larger and/or interdisciplinary scientific projects are being tackled. While the project groups share the same goals, they may not share the same research tools. As part of a multi-group collaboration, a lab must be able to prepare data in multiple formats to fit the needs of multiple collaborators.
R4	*Automate data manipulation*. Automation removes the burden of repetitive activities from users and helps minimize human errors. In addition, scaling up is much more feasible for an automated system than a manual one. When preparing data for collaborations, automated data manipulation protocols can also be used to facilitate internal data processing. Having a single mechanism for initiating automated processes can simplify overall data management for users.
R5	*Control data access*. Data sharing must not usurp the data owners' ability to control how data is published. In particular, sensitive data, e.g., patient information, are often used in clinical research but must not be shared with all collaborators without careful consideration and adherence to applicable regulations or restrictions. This implies that data owners are not forced to store their data in a centralized repository or in a prescribed format.

3. The PMT Project

The UCSF Pharmacogenetics of Membrane Transporters project (PMT, http://pharmacogenetics.ucsf.edu) [5] provides an excellent example of the type of collaborative science project that can benefit from the DASH infrastructure. The goal of the PMT project is to understand the genetic basis for variation in drug response for drugs that interact with membrane transport proteins. Membrane transporters, a major determinant of pharmacokinetics, are of great pharmacological importance. The UCSF PMT project, begun in April 2000, involves more than 50 researchers from diverse disciplines, distributed across 19 labs at UCSF, UCLA, San Francisco General Hospital, and Kaiser Foundation Hospitals. These investigators are systematically identifying sequence variants in transporters and determining the functional significance of these variants through evaluation of relevant cellular and clinical phenotypes. Experimental results are deposited in the Pharmacogenetics Research Network and Knowledge Base (PharmGKB, http://www.pharmgkb.org) [6], hosted at Stanford University. The PMT is organized into four major components:

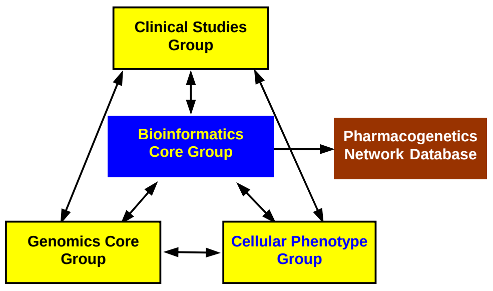


Figure 1. Components of the Pharmacogenetics of Membrane Transporters project.

- Genomics Core Group (GCG), which is sequencing the DNA of 200 genes from several sample sets of more than 250 individuals;
- Cellular Phenotype Group (CPG), which determines the pharmacological effects of single nucleotide polymorphisms (SNPs) in cellular assays;
- Clinical Studies Group (CSG), which tests drug response in volunteer human subjects with known genotypes; and
- Bioinformatics Core Group (BCG), which performs data analyses, provides computing infrastructure to facilitate information exchange, and exports results to the PharmGKB.

Figure 1 shows the flow of data within the PMT. GCG-BCG data exchanges include trace files from DNA sequencers, per-sample single nucleotide polymorphism sites and variants. Data from GCG to BCG are uploaded to a shared network file system in Common Assembly File and

Standard Chromatogram File formats. BCG analyzes the uploaded experimental data and makes the results available through the PMT intranet Web site (password protected), either as Hypertext Markup Language (HTML) pages or as a tab-separated-values plain text file suitable for importing into spreadsheets and databases. With over 50 sequencing experiments completed and 150 proposed for the next five years, the timely analysis of data sets is critical to the project. In addition to standard analyses such as Hardy-Weinberg equilibrium of SNPs, other more speculative analyses are constantly being proposed and tested. Leabman has recently published some of these "data mining" analyses [7]. While the BCG has kept pace with the current volume of PMT data primarily through use of manual approaches to data curation, automation tools are needed to handle the planned increase in volume of data, as well as new hypotheses-testing analyses, for the next five years.

4. Existing Technologies

Tools exist to address some of the requirements listed in Table 1. The four most relevant technologies are workflow management, data integration, distributed resource management, and event-based technologies.

Workflow management is a protocol-centric paradigm for controlling activities within a system; data is often treated as auxiliary information attached to process instances. There are many commercial [8-10] and open-source [11-13] workflow management systems. Most provide central management of workflows, where an analysis produces the initial workflow, which is then revised under strict access control. However, the classic workflow paradigm does not fit well in a collaborative science environment, where many researchers need to be able to introduce new data and activities into a process definition. Centralized control over workflow modification would introduce unacceptable overhead (i.e., does not address requirement R2 in Table 1). Additionally, many commercial workflow systems are devoted to streamlining the execution of a series of manual activities. However, in a research environment workflows are often data-driven or data-triggered, and updates can be handled by automated activities rather than manually. With their focus on manual activities, traditional workflow systems often do not handle requirement R4 gracefully.

We have investigated both commercial and open-source workflow solutions. The system that most nearly satisfies the requirements in Table 1 is myGrid [11]. myGrid emphasizes the large-scale, geographically distributed e-science environment, and is necessarily complex. The subset of myGrid's functionality most pertinent to our requirements is limited to three of these components: an information repository, workflow enactor, and notification service. However, there does not appear to be the level of functional integration between these components necessary to support the kind of features outlined in our requirements. While myGrid seems like a promising technology, it appears to be overly complex for use in simple collaborations.

Data integration is data-centric; processing activities are not explicitly included other than as clients that access the integrated data views. Thus, requirements R1 and R2 are often not well addressed by data integration products. It is also unclear whether the benefits of data integration outweigh its cost. The volume of exchanged data in a research collaboration is typically low compared with enterprise-level data stores. Frequently, a simple data transfer and processing strategy is sufficient and does not incur the overhead of creating and managing an integrated data view between collaborators.

Distributed Resource Management solutions, or DRMs, include workload balancers and batch management systems like OpenPBS [14] and Platform Computing's Load Sharing Facility (LSF) [15], as well as grid solutions such as the Globus Grid Toolkit [16]. While these systems are very useful for the utilization of computational resources, they do not provide the capability to define pipelines based on data availability or modification. These systems could be used as part of a pipeline to distribute the computational task across multiple nodes, but do not themselves meet the requirements discussed above.

Event-based technologies are commonly used in conjunction with user interfaces and user-oriented systems [17, 18], but event-based approaches have also been used for distributed systems [19-21]. All of these systems function in a similar manner: events are generated in response to some action and are processed by an event handler for that particular event. Event handlers may generate additional events or might update files, database tables, or a user's display. Event-based approaches provide a firm foundation for building computational pipelines by linking together various events, but with the exception of Metis [22] have not been widely utilized for that purpose. Many of the requirements defined in Table 1 are very data-centric, and it may not be apparent to users how these can be met by the finer granularity event-based approach. However, it should be noted that this approach offers a great deal of promise due to its flexibility and adaptability.

While none of these approaches independently addresses all the requirements in Table 1, based on our analysis we felt that designing DASH using an event-based model offered the best tradeoff between functionality and usability. Our approach to ameliorate the complexities of the event model was to layer a data flow representation on top of the more granular event model.

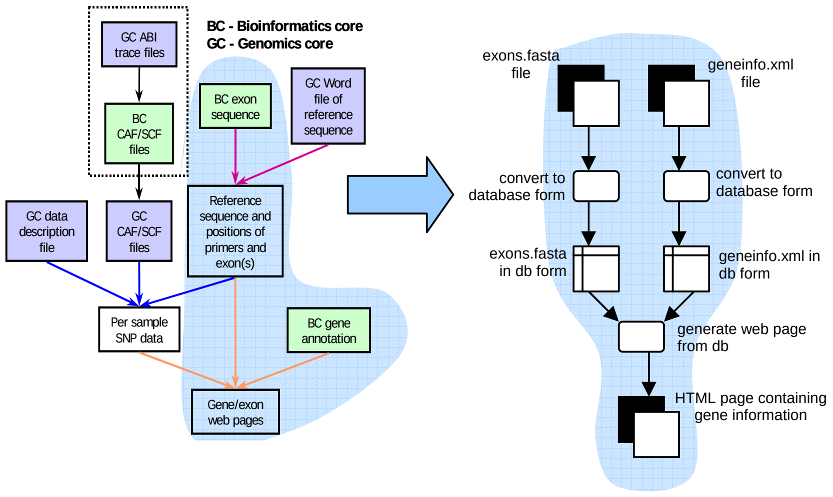


Figure 2. Illustration of the pipeline constructed to automate data processing within the Bioinformatics Core component of the PMT project. The shaded section on the left represents the portion of PMT data processing encapsulated by the data flow diagram shown on the right. This data flow is then used to automate the generation of web pages from source data.

5. Workflow within the PMT

The diagram on the left in Figure 2 is a high-level representation of the workflow between two of the components in the PMT project. (See reference [5] for a detailed description.) The PMT, with its complex interactions between heterogeneous, distributed resources, well represents the many small- to medium-sized collaborative science projects for which we have designed DASH. In Figure 2, we identify a subsection (shaded) of the PMT processing pipeline from which we isolated a simple data processing activity that generates web pages from data files. From this existing process we extracted a simple, two-staged pipeline. The image on the right illustrates this pipeline utilizing a standard set of data flow diagram symbols adapted from the Gane and Sarson method of process notation [23]. Using this PMT pipeline as an example, we implemented a simple proof-of-concept prototype of DASH that monitors a file or database table and triggers a protocol to run whenever the associated data source changes. As depicted in Figure 2, this two-staged pipeline creates a web page containing gene information anytime a new *exons.fasta* or *geneinfo.xml* file appears.

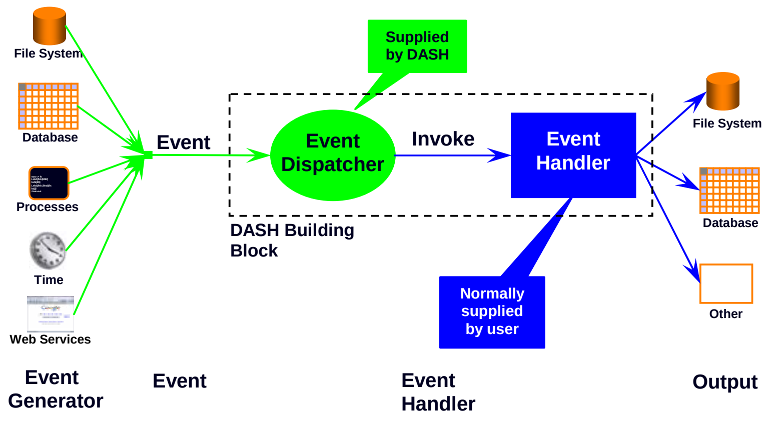


Figure 3. Overview of components in DASH's event model.

6. The DASH Event Model

Science can be viewed as an event-driven process; the development of a new hypothesis, availability of new data in a public database, or generation of experimental results are all examples of important events in the scientific research environment. Some of these events originate from human actors such as fellow researchers, while others occur as the result of automated processes. Each one of these events may warrant any number of follow-up actions: experiments may need to be designed to test the hypothesis, automated protocols could be invoked to process the novel data, or the laboratory results could be compared against previous experiments. Note that any of these activities could, in turn, generate more events that will require additional processing, and so on. In general terms, this process consists of three core concepts: actions that generate events, the events themselves and attached data, and handlers that process events.

Even the simple pipeline shown in Figure 2 evokes some of the potential complexity of automated data monitoring and processing. The two input branches of the prototype pipeline run independently, each updating its own set of database tables. An update to either table triggers the web page generation protocol with sufficient data to produce a web page. The two inputs can be thought of as having a Boolean OR relationship. Within the PMT, this is not always the case; some data sources have a Boolean AND relationship. That is, a step in the processing pipeline requires updated data from two or more data sources before it can run. Therefore, a useful data automation system must also handle a group of data sources having an AND relationship. These relationships suggest the concept of data groups as opposed to individual sources. Taken to

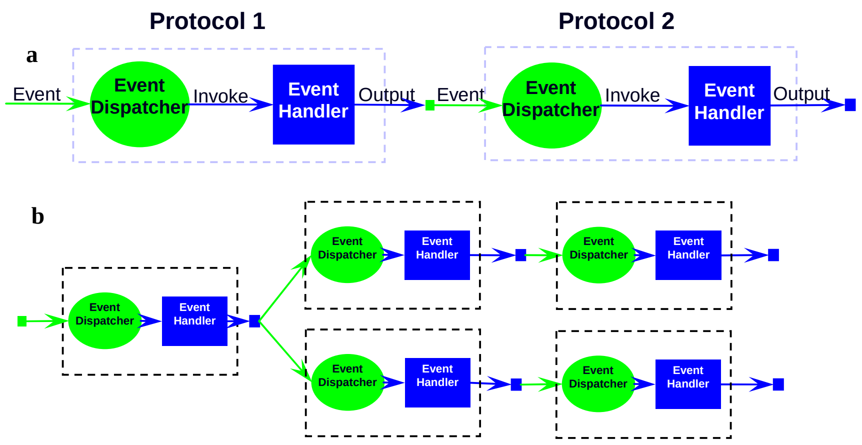


Figure 4. (a) Building blocks of DASH's event model; (b) Several building blocks linked together form a complex event processing pipeline.

the next level, this analysis implies the requirement of nested groups to support arbitrarily complex data processing scenarios.

Figure 3 shows an overview of all of the components in the DASH event model. The Event Dispatcher is a component of DASH that monitors Event Generators for Events and subsequently dispatches these Events to their respective handlers. The dotted line labeled "DASH Building Block" encompasses one processing unit of the DASH event management architecture.

Figure 4a shows two event processing units. The role of an individual event-processing unit is simple; an event arrives from DASH's event dispatcher, and the handler is invoked. This unit need not have any knowledge of where the event came from, which other handlers may process the same event, or what types of events may be generated as the result of the event handler invocation. This restricted local view keeps the conceptual model simple, while providing a high level of flexibility and applicability to known and unforeseen problem domains.

The real power in this model comes from associating these simple event-processing units. Associations between individual units are implicitly established when events generated from the action of one unit's event handler are dispatched to the event handler of the second. DASH provides a registration interface that allows users to specify what types of events each Event Generator is capable of producing, as well as which Events should be dispatched to which Handlers. The linking of processing units can result in complex relationships between multiple event generators and handlers, as illustrated by the branching pipeline in Figure 4b.

7. The DASH Data Flow Layer

While events and handlers provide a sound foundation upon which to build, from the researcher's perspective it is advantageous to view the interactions in terms of the more familiar concepts of data and processing protocols; these can be represented using data flow notation. Figure 5a introduces a new view of the DASH building block using data flow notation. Conceptually similar to the event model building block introduced in Figure 3, this construct represents the functional unit in the context of data flow. D1 represents a data store and P1 a protocol that is invoked in response to changes in that data store. Within this simple system, there is only one execution pattern: D1 is altered and P1 is invoked to process the changed data.

These building blocks can be aggregated to form complex pipelines of data and processing protocols, as shown in Figure 5b. While representing relationships in terms of data and protocols (as opposed to events, generators, and handlers) is more suitable for a data sharing application, the underlying implementation still uses the event model to propagate changes throughout the system. A change in data store E1 generates a data change event. Protocols P1 and P2 are registered as handlers for data change events in E1, and are invoked by the event dispatcher to process the changed data. Similarly, the actions taken by protocols P1 and P2 (e.g. writing information into a database table) could result in the generation of further data change events from data stores D1 and D2, respectively. This process of event generation and event consumption thus

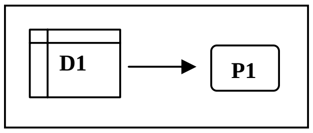


Figure 5a. Data flow building block.

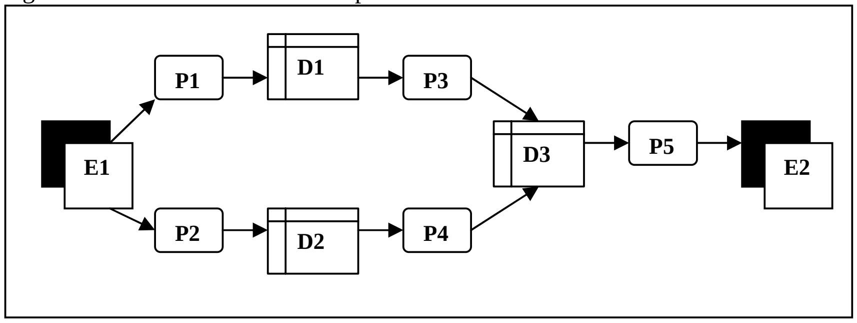


Figure 5b. Complex data flow diagram using building blocks.

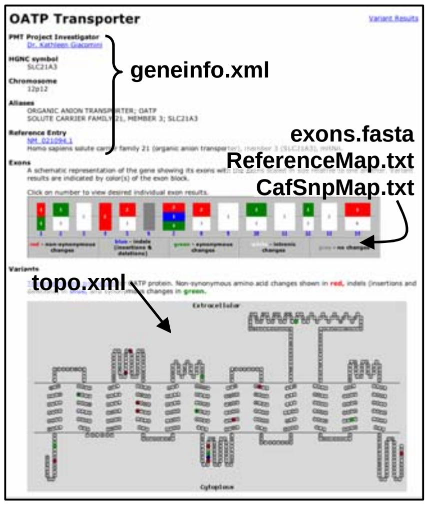
drives the propagation of data updates throughout a processing pipeline.

In the context of data flow, it is sometimes advantageous to process events in an order different from that in which they are generated. Reordering event processing enables DASH to maintain data integrity while optimizing use of resources such as data access and processor time. The end result of this event-processing model is to propagate all data updates throughout the system in the most efficient manner possible while still maintaining an internally consistent set of data. Events can be combined into a series which can then be optimized to allow for the efficient processing of large volumes of data by running protocols in parallel whenever possible.

Our DASH infrastructure, consisting of an underlying event model that is optimized for efficiency and ease of use in a data-sharing context, addresses several of the requirements given in Table 1. Using a preexisting body of distributed data stores (R5) and heterogeneous processing protocols, DASH can automate the propagation of updates through the system in order to provide a consistent set of data for use in subsequent processing steps (R4), or for consumption by collaborators or end-users (R3).

8. DASH Applied to the PMT

Figure 6 shows a page from the PMT Website generated using an expanded version of the prototype described in section 5 above. The text accompanying the arrows gives the source files for the data. The data used to generate this single Web page comes from at least five different files—some of them in XML format, others in fasta format, and still others in plain text. Changes in any of the source files will result in the automatic regeneration of the Web page by DASH.


OATP Transporter

Variant Results

PMT Project Investigator
Dr. Kathleen Giacomini

HGNC symbol
SLC21A3

Chromosome
12p12

Aliases
ORGANIC ANION TRANSPORTER; OATP
SOLUTE CARRIER FAMILY 21, MEMBER 3; SLC21A3

Reference Entry
NM_021094.1
Homo sapiens solute carrier family 21 (organic anion transporter), member 3 (SLC21A3), mRNA.

Exons
A schematic representation of the gene showing its exons with [illegible]. Variant results are indicated by color(s) of the exon block.

Click on number to view desired individual exon results.

red - non-synonymous changes | blue - indels (insertions & deletions) | green - synonymous changes | white - intronic changes | grey - no changes

Variants
[illegible] OATP protein. Non-synonymous amino acid changes shown in red, indels (insertions and deletions) in blue, and synonymous changes in green.

Figure 6. Sample PMT web page containing data about the exons of a transporter gene and showing the data files used to generate the HTML representation.

9. Current Status

DASH is currently in the early development stage. A simple prototype has been implemented and applied to the PMT project to obtain the results described in this manuscript, and further implementation work is ongoing. As DASH becomes more functional, our intent is to make the software available at no cost and in documented source code form so that other research groups can directly benefit from our work. Further information on DASH is available from http://www.cgl.ucsf.edu/Research/DASH/.

10. Future Work

There are two areas that we will be focusing on for our future implementations of DASH. The first of these is the development of a Web-based user interface that will allow researchers to create, modify, and monitor DASH processing pipelines. By enabling researchers to graphically manipulate active processing pipeline components, we can address requirement R2. The ability to discover, display, and print relationships among the processing pipelines of collaborating researchers allows us to address requirement R1.

The second area for future work is the extension of DASH to cross organizational boundaries. This often requires that the tools and libraries support some form of distributed computing model. For DASH, distributed computing will be supported through Web Services [24] interfaces. The use of Web Services allows us to leverage the security mechanisms already supported through web services technologies and provides a general interface between two DASH instances running in different computing environments. Web Services will also be used to support communication between DASH and other related systems that support a Web Services interface. DASH will act as a Web Services endpoint as well as a Web Services client. One additional use of Web Services will be to export administrative information for the discovery and status of DASH pipelines (bounded, of course, by security restrictions). This will be used to present a broader view of processing pipelines across multiple computing environments (addresses requirements R3 and R5).

Acknowledgments

The UCSF Pharmacogenetics of Membrane Transporters project is sponsored by NIH grant U01 GM61390. Support for the DASH project comes from NIH P41-RR01081.

References

1. Models of Team Science (www.becon1.nih.gov/symposia_2003/BECON2003_sessionIV.ppt).
2. BECON 2003 Symposium on Catalyzing Team Science (www.becon1.nih.gov/symposium2003.htm).

3. Zerhouni, E., *The NIH Roadmap.* Science, 2003. **203**: p. 63-64,72.
4. NIH Roadmap Initiatives (http://nihroadmap.nih.gov/initiatives.asp).
5. Stryke, D., et al. *SNP analysis and presentation in the Pharmacogenetics of Membrane Transporters Project*. in *Pacific Symposium Biocomputing*. 2003: World Scientific.
6. Hewett, M., et al., *PharmGKB: the Pharmacogenetics Knowledge Base.* Nucleic Acids Res., 2002. **30**(1): p. 163-165.
7. Leabman, M.K., et al., *Natural variation in human membrane transporter genes reveals evolutionary and functional constraints.* Proc Natl Acad Sci U S A, 2003. **100**(10): p. 5896-901.
8. IBM Lotus Workflow (http://www.lotus.com/).
9. Oracle Workflow 11i (http://www.oracle.com/).
10. BEA Weblogic Integration (http://www.bea.com/framework.jsp).
11. Stevens, R.D., A.J. Robinson, and C.A. Goble, *myGrid: personalised bioinformatics on the information grid.* Bioinformatics, 2003. **19 Suppl 1**: p. I302-I304.
12. Oinn, T., et al., *Taverna: a tool for the composition and enactment of bioinformatics workflows.* Bioinformatics, 2004.
13. Vivtek Inc., *wftk (Workflow Toolkit).* 2003.
14. Portable Batch System (http://www.openpbs.org).
15. Platform LSF (http://www.platform.com/products/LSF/).
16. Foster, I. and C. Kesselman, *Globus: A Metacomputing Infrastructure Toolkit.* International Journal of Supercomputer Applications, 1997. **11**(2): p. 115-128.
17. Document Object Model (DOM) Level 2 Events Specification (http://www.w3.org/TR/2000/REC-DOM-Level-2-Events-20001113/).
18. Jacob, R.J.K., L. Deligiannidis, and S. Morrison, *A software model and specification language for non-WIMP user interfaces.* ACM Transactions on Computer-Human Interaction (TOCHI), 1999. **6**(1).
19. Carzaniga, A., D.S. Rosenblum, and A.L. Wolf, *Design and Evaluation of a Wide-Area Event Notification Service.* ACM Transactions on Computer Systems, 2001. **19**(3): p. 332-383.
20. Cugola, G., E. Di Nitto, and A. Fuggetta. *Exploiting an event-based infrastructure to develop complex distributed systems*. in *20th international conference on Software engineering*. 1998.
21. Ben-Shaul, I.Z. and G.E. Kaiser. *A paradigm for decentralized process modeling and its realization in the Oz environment*. in *16th international conference on Software engineering*. 1994.
22. Anderson, K.M., et al. *Metis: Lightweight, Flexible, and Web-based Workflow Services for Digital Libraries*. in *2003 Joint Conference on Digital Libraries*. 2003. Houston, Texas: IEEE.
23. Gane, C.P. and T. Sarson, *Structured Systems Analysis: Tools and Techniques*. 1979: Prentice Hall. 241.
24. Web Services Activity (http://www.w3.org/2002/ws/).