Toward Automating the Development of Grid Applications in Bioinformatics

Xiujun Gong[1], Kensuke Nakamura[2], Kei Yura[2], Nobuhiro Go[1, 3]

[1]Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5 Takayama, Ikoma, Nara, 630-0192 Japan

[2]Center for Computational Science and Engineering, Japan Atomic Energy Agency, 8-1 Kizu, Souraku, Kyoto, 619-0215 Japan

[3]Neutron Science Research Center, Japan Atomic Energy Agency, 8-1 Kizu, Souraku, Kyoto, 619-0215 Japan

Abstract

In silico bioinformatics experiments involve integration of and access to computational tools and biological databases. The emerging grid computing technologies enable bioinformatics scientists to conduct their researches in a virtual laboratory, in which they share public databases, computational tools as well as their analysis workflows. However, the development of grid applications is still a nightmare for general bioinformatics scientists, due to the lack of grid programming environments, standards and high-level services. Here, we present a system, which we named Bioinformatics: Ask Any Questions (BAAQ), to automate this development procedure as much as possible. BAAQ allows scientists to store and manage remote biological data and program resources, to build analysis workflows that integrate these resources seamlessly, and to discover knowledge from available resources. This paper addresses two issues in building grid applications in bioinformatics: how to smoothly compose an analysis workflow using heterogeneous resources and how to efficiently discover and re-use resources available in the grid community. Correspondingly an intelligent grid programming environment and an active solution recommendation service are proposed.

Keywords: Active Solution Recommendation, Bioinformatics, Grid Application, Task Mapping Editor.

1 Introduction

In silico bioinformatics experiments involve integration of and access to computational tools and biological databases(Wroe, Goble, Greenwood, Lord, Miles, Papay, Payne and Moreau 2004). Bioinformatics scientists need to orchestrate a growing number of these resources to perform their analysis(Blythe, Deelman and Gil 2004; Cannataro, Comito, Schiavo and Veltri 2004; Gil, Deelman, Blythe, Kesselman and Tangmunarunkit 2004). Bioinformatics is a collaborative discipline, in which bioinformatics scientists need to share their ideas and analysis method not only through manuscripts, but also through information repositories, such as public databases, web services and analysis workflows(Buetow 2005). The emerging grid technology(Foster, Kesselman and Tuecke 2001), whose initiative is to enable scientists of similar interests to conduct their researches in a virtual organization, is forming the base of the new generation collaboration platform.

Although numerous grid middleware are being made available(Imamura, Yamagishi, Takemiya, Hasegawa, Higuchi and Nakajima 2003; Rowe, Kalaitzopoulos, Osmond, Ghanem and Guo 2003; Sulistio, Poduvaly, Buyya and Tham 2005), the development of grid applications is still a nightmare for general bioinformatics scientists, due to the lack of grid programming environments, standards and high-level services. To ease the use of available resources without concerning the low level details of how the individual grid components operate, many researches have focused on studying high level services. Asia Pacific BioGrid (*http://www.apbionet.org/grid/*) project is trying to build a customized, self-installing version of the Globus Toolkit, a distributed environment for designing and managing grid. It comprises well-tested installation scripts and avoids dealing with Globus details. Bio-Grid working group(Pytlinski, Skorwider, Bala, Nazaruk and Wawruch 2002) is developing an access portal for modeling biomolecular resources. The project develops various interfaces for biomolecular applications and databases that will allow biologists and chemists to submit works to high performance computing facilities, hiding grid programming details. As one of the United Kingdom e-Science projects, myGrid(Stevens, Robinson and Goble 2003; Miles, Papay, Wroe, Lord, Goble and Moreau 2004) has been developed as an open source data-intensive bioinformatics application on the grid. Data integration is achieved both by dynamic distributed query processing and by creating virtual databases through federations of local databases. To enhance access to bioinformatics resources, Pegasys(Shah, Sawkins, Druce, Quon, Lett, Zheng, Xu and Ouellette 2004) is designed to map an abstract workflow, which is composed using ontology-driven approach, onto the grid. However, most of the projects above put more efforts on transparently accessing to a single resource, little on how to connect them to form a meaningful workflow and how to discover knowledge from existed workflows for building new analysis.

In this paper, we present a system, which we named "Bioinformatics: Ask Any Questions (BAAQ)". BAAQ allows scientists to access and manage biological databases and computation tools in the grid, and to ease building analysis workflows. The system is also capable of managing and sharing these workflows, and discovering knowledge from available resources. Specifically, BAAQ has the following features:

- It is a flexible, grid-based infrastructure that allows integration of data and computational intensive applications;

- It provides an intelligent grid programming environment that eases the works of composing analysis;
- It provides a search engine that recommends workflow-based solution candidates considered as references or parts of new analysis for users' requirements.

The reminder of this paper is organized as follows: Section 2 describes the system architecture forming functionality of BAAQ; Section 3 presents an intelligent grid programming environment; Section 4 introduces an active solution recommendation service; and section 5 concludes this paper and discusses future works.

2 System architecture

The components of BAAQ can be divided into four groups: communication infrastructure group, information service group, service assistant group and application resource group, as shown in Fig 1.

- The communication infrastructure group provides a grid-based computational environment that is responsible for authentication, authorization, communication and computational resource location and allocation. Its implementation is based on Seamless Thinking Aid (STA) (Takemiya, Imamura and Koide 1999), which is IT-based Laboratory (ITBL) environment for assisting parallel programming. STA uses nexus developed at Argonne National Laboratory as communication library and can employ diverse communication protocols such as TCP/IP, AAL5 and MPL. STA provides a file manager through which users manage remote files as if they are in a single machine. The tool manager of STA allows managing and integrating grid-based middleware.
- The information service group consists of grid middleware such as Task Mapping Editor (TME) (Imamura, Yamagishi, Takemiya, Hasegawa, Higuchi and Nakajima 2003), Active Solution Recommendation (ASR), TextBrowser, and PluginTool. These middleware modules interact with users through a GUI interface and communicate with STA by corresponding adaptors. TME is the core component for managing resources and building workflows. ASR recommends solution candidates by looking up existing resources. TextBrowser is used to browse the text-based content of data resources. PluginTool is designed to visualize the content of data resources using client side software.
- The assistant service group provides a set of Application Program Interfaces (APIs) to the service group for helping accessing and managing application resources. Metadata management module is designed to collect and maintain a metadata repository which contains information about available resources and system configuration. Filter management module is responsible for manipulating filters that mediates data formats. Wrapped Program Toolkits (WPT) is used to help users encapsulate bioinformatics programs with uniform interfaces by utilizing filter repository. Active Service Provider (ASP) has the following three functions: 1) provide guidelines for choosing a resource and a set of parameters for program resources; 2) aid connecting resources in the process of workflow generation of TME; and 3) recommend a visualization tool (such as TextBrowser or PluginTool) for viewing analysis results.
- The application resource group consists of data and program distributed on heterogeneous computers and their descriptions, as well as workflows that describe how data and programs are connected to perform a certain analysis. Hereafter, we call these three kinds of resources as bio-resources. The information service group accesses these resources by the help of service assistants.

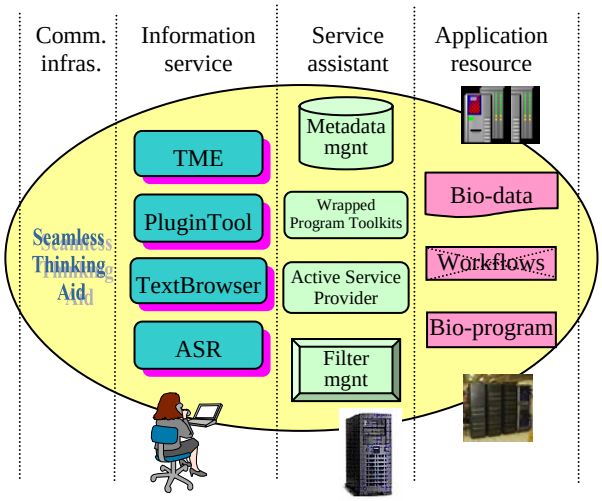

Fig 1 Architecture of BAAQ

In this architecture, the biological data and analysis programs (and their replicas) distributed on different computation nodes are registered through TME. These resources are encoded with knowledge rich metadata and are stored under the tree-like structure of TME workspaces. Before registering the analysis tools, they are wrapped so that they have uniform interfaces. To compose an analysis workflow, a user only needs to select corresponding resources and link them by drawing a line between them. ASP is helpful for this procedure as described above. The analysis results are usually stored as remote files, managed as icons in the TME workspace, and can be viewed using customized visualization tools such as TextBrowser and PluginTool. An alternative way for building a new analysis is to use ASR. ASR recommends solution candidates by looking up available resources. Once a user finds candidates relevant to his/her requirements, one can import them based on a certain proxy and reconfigure their parameters to meet his/her specific needs.

3 An intelligent grid programming environment

Task Mapping Editor (TME) (Imamura, Yamagishi, Takemiya, Hasegawa, Higuchi and Nakajima 2003), one of the components of STA, is a visual programming tool developed at the Center for Computational Science and

Engineering at Japan Atomic Energy Agency. TME has been developed for handling distributed resources and supporting the integration of distributed applications. All the resources are represented as icons on TME workspaces and data dependency is defined by a line linking the icons. Using TME, a user can design a workflow diagram of the distributed applications, just like using a drawing tool. We equip TME with the following functions to handle bioinformatics problems:

3.1 Scalable filter library

Data format transformation is a tedious work for bioinformatics scientists; hence it is becoming an active research area in bioinformatics. Two typical methods are data warehousing and federation, both of which adopt SQL-like languages as interfaces with end users(Lacroix 2002). Our method, data filter approach, is quite similar to the federation method in principle. Instead of providing SQL interfaces, we use the filter manager [Fig 2] to call corresponding filters to perform intended transformation.

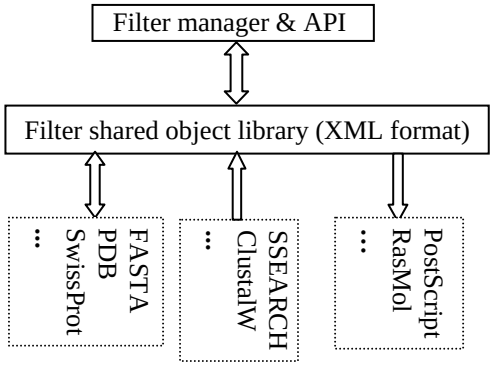

Fig 2 filter repository structure

We define a uniform XML schema with enough capability of describing biological data sources and typical analysis results of bioinformatics tools. To exchange different data formats with the defined XML format, we design a set of filters (Fig 2). Upon a user's requirement, the filter manager accesses corresponding filters to perform intended transformation. Using this filter library, the data filter service can transform formats between any two different data sources. One of the advantages of this architecture is that developers only need to change the individual filter when a data source is changed. This filter library also contains two other classes of filters. One is the filter that extracts some results from program outputs into XML format and is used to wrap programs. The other is to convert some XML analysis results into PostScript format or RasMol script format and is used in the visualization module. Each filter has a replica on each computation node. The system will select an appropriate one at the time of workflow execution.

3.2 Active service provider

TME has already provided abundant utilities for composing and managing a workflow with distributed resources. Each resource is represented as an icon. Building a connection between different resources is just to draw a line between icons. For specific application areas, however there are still some problems: 1) What data and programs should a user choose to perform a specific bioinformatics analysis? 2) How does a user keep semantic consistency when building a connection between two icons? For example what would happen if a user connects two icons with heterogeneous formats?

For the first problem, we provide a tree-based browser for biological databases and tools. In TME window, each data or program resource is represented as an icon under the corresponding taxonomy tree. A user can search for the target objects by traveling through the taxonomy tree. Also one can use ASR discussed in the next section to search for the available resources by a natural language interface.

For the second problem, although TME can check some syntactic errors for the whole workflow, little emphasis is put on semantics. We provide a method, called Active Service Provider, for solving the problem. Our Active Service Provider includes two procedures: semantic check and service provider. When a user connects two icons, Active Service Provider will be triggered to check whether this connection is admissible by examining the corresponding profile, which is an XML formatted file to describe resource's attributes, such as public interfaces, accessibility and functionality. If allowed, the system will detect where extra services are needed. For example, if a user wants to connect two icons with different data formats, the data filter service will be provided to perform data format transformation.

3.3 Wrapped Program Toolkits

Program integration is a very complex activity due to, for example, the heterogeneity of manipulated data and incompatible interfaces. In most cases, programs are black-box systems that do not allow any change inside the programs.

TME organizes a program as an icon with configurable interfaces. Its function and semantics of its interfaces are still hidden from users. Users have to shift more efforts for studying the function and understanding the meaning of each interface. Most bioinformatics tools are developed by different researchers or commercial units. The data formats that programs can accept are quite different from one another. In BAAQ, each program is associated with two descriptive files: profile and configuration. The profile encodes three kinds of information: 1) source information about the program, for example the URL, from which the program is downloaded, version, release number, and code types(C/C++, Perl, Fortran and so on), is used to maintain consistence of different versions; 2) function summary information, for example analysis types (sequence alignment, structure prediction), methods and performance evaluation, function description, for human understanding or retrieval purpose; and 3) function and interface description of each program for semantic check. The configuration file ensures that programs in the package are executable on different platforms (Linux, SunOS and IRIX and so on).

Program interfaces play a particular role in composing analysis workflows. Heterogeneity and diversity in program interfaces exhausts most users. To solve this problem, we classify the interfaces into three types: switch,

real-value and file parameters. We can list all possible values for the switch parameters and limit the maximum and minimum for the real-value parameters. For the file parameters, the situation becomes a little more complex, because of the diversity in biological data formats. Our solution is to unify data formats of the file parameters as XML formats as much as possible. Since XML is more flexible to develop machine understandable code, we encode related information into XML files so that the system can identify them automatically. To this end, we design many filters that extract related information from the original outputs of the analysis tools into XML files [Fig 2]. Using these filters, the programs are wrapped easily.

3.4 Visualization for analysis procedure / result

Visualization plays a vital role in understanding the analysis procedures and results. Using the graphically-enabled task editor environment in TME, a user can visually design and view the workflow, monitor its execution status and trigger corresponding tools to browse the content of an individual icon in the workflow. In the visualization module, we focus on the visualization of analysis results through triggering the third party visualization tools. There is a large amount of software for presenting biological data graphically, for example, Protein Explorer, RasMol, Chime for protein structure, gsview and GNU gv for many kinds of printable pictures and GnuPlot and SciLab for statistical data. To integrate those kinds of programs, we face two challenging problems: 1) Although most of them have different versions working on different platforms, for example RasMol can run on Windows, Mac and Unix-based platform, few of them are platform-independent; and 2) There are many programs for visualizing the biological data in a similar way. However, users usually prefer the one they are most accustomed to.

In BAAQ, users can use the programs hosted in client side by the way of Plugin, which most web browsers provide. What the system does is to provide targeted data and to recommend content type, by which the browser decides which program on the client side will be triggered. Once content type is identified, Active Service Provider will call the corresponding filter (Fig 2) to transform the XML file into the corresponding format (i.e. PostScript, RasMol), then trigger the PluginTool to view the data. In this way, users can customize their preferred visualization tools.

4 An active solution recommendation service

In the previous section, we addressed the issue how the bio-resources are glued together to compose an analysis workflow. Issues we address here include how to choose the available bio-data and program resources and how to discover and reuse existing workflows for the developers of grid applications. We believe that existing workflows enclose abundant knowledge. By investigating these workflows, we can, not only learn how bio-resources are used, but also know how the analysis results are derived. In some sense, knowing how the result is derived is more important than the result itself. We propose a prototype called Active Solution Recommendation (ASR) to solve these problems. As the name suggests, the goals of ASR are to discover the bio-resources, to allow the reuse of the most relevant workflows, and to recommend solution candidates for user's questions.

The implementation of ASR is based on mature web searching and data mining technologies. But it distinguishes the traditional search engines by two aspects:

- Compared with the great diversity of web contents, its sources are well organized, so that it allows more efficient search algorithms.
- ASR is not only responsible for finding the targeted resources, but also provides a set of utilities for manipulating the candidates to be shared and reused in the grid environment. However, the concern of a general web search engines focuses on how to locate the sources. It is a user's task to find out how to use them.

The overall architecture of ASR is shown in Fig 3. Its main modules are Information Crawler, Indexer and Query/Answer Analyzer.

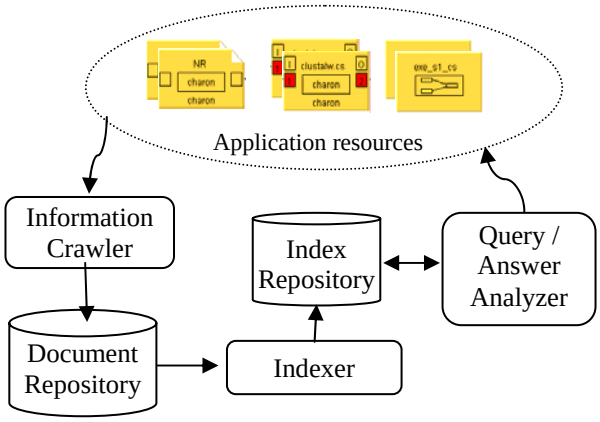

Fig 3 the architecture of ASR

4.1 Information Crawler module

The Information Crawler module is responsible for automatically retrieving bio-resources into a central repository. In BAAQ, each bio-resource is described by an XML file and associated with a Uniform Resource Identifier (URI). And each bio-data or program resource binds to at least a source identified by a Uniform Source Identifier (USI). The system will select a suitable USI at the runtime of a workflow. A workflow links other resources together for performing an analysis. Intuitively, if a bio-resource is referred frequently, probably it is more significant than others. Thus, the linking information here plays the role as a hyperlink does in traditional search engines, such as Google. The system also allows users to make notes based on their opinions about bio-resources' capabilities. The Information Crawler collects information above into a document repository.

4.2 Indexer module

The Indexer module is responsible for building and maintaining the index data structure of ASR. This step is very important, because some information about the relevance of bio-resources within the grid must be

integrated and indexed for efficient search and discovery. The information we consider to be important includes:

- functional information;
- annotation information;
- linking information.

The functional information provides a more complex but precise description of functionality offered by the bio-resources. For instance, the functional information of a bio-data resource describes the characteristic of bio-data such as data format, published method, and a simple abstract showing how the data is generated. The description of bio-program resource includes its published method, copywriter information, classification in TME workspace and a profile showing how the program works. For the case of a workflow, the situation becomes more complex due to the complexity in its components. We propose a workflow description framework through using a controlled vocabulary, the similar idea which has been used to describe experimental procedures and common process in the Human Proteome Organization Protein Standards Initiative (Orchard, Montecchi-Palazzi, Hermjakob and Apweiler 2005). The organization of our vocabulary follows a hierarchical structure. It provides a clear and comprehensive functional description of a workflow.

The annotation information involves the evaluation from different users who have obtained and used the resource.

The linking information regards that how the resource is referred by other resources and that who have visited the resource. All the information is used to provide the significance of its quality.

The Indexer builds an index repository by parsing and weighting the information above.

4.3 Query/Answer Analyzer module

The third module of ASR is the Query/Answer Analyzer, which actually solves a user's requirement based on the index repository that has been built in the Indexer module. This module takes a user's query as well as his/her preference, which has been captured from his/her query history, as input, and outputs a ranked list of candidate answers. The system also provides a set of utilities for manipulating the results. Specifically, the services provided include:

- annotation service to enable a user to evaluate the candidate from the user's view;
- reservation service to enable a user to make a reservation of the candidate so that she/he can import it;
- authorization service to enable a user to authorize a candidate that has been reserved so that other users can import it;
- import/export services to enable a user to import an authorized candidate and to export a reserved one. The import/export service is of great importance in the sense of collaborations among users, because this step makes resources really shareable and reusable.

Once a user imports a workflow, the system will deploy all the components automatically. Also he/she can change their parameters or use it as a part of new analysis to meet his/her specific needs.

5 Conclusion and future directions

In this paper, we have presented an integrated framework for building grid application in bioinformatics. In a grid-based environment, we developed an intelligent grid programming environment for composing an analysis workflow smoothly that integrates distributed and heterogeneous resources. Another contribution of this paper is the development of Active Solution Recommendation service, which allows users to search for and reuse the bio-resources they queried. This service is based on mature web search technologies and provides a set of utilities for manipulating results so that users can share and exchange their analysis. We believe that in the near future there will be a growing demand for this kind of service.

One of our future works would be to incorporate the ontology approach, which have been applied successfully in many bioinformatics applications(Bakera, Brassa, Bechhoferb, Gobleb, Patonb and Stevensb 2000; Lacroix 2002), into our framework. Clearly, an appropriate ontology is helpful for both the uniform access to heterogonous databases and the semantic description of bio-resources. Another future work is to refine the solution candidates of ASR. Ideally we expect that ASR can generate a new meaningful analysis for a user's query.

6 Acknowledgement

This work was supported by the Special Coordination Funds Promoting Science and Technology from Ministry of Education, Culture, Sports, Science and Technology (MEXT) of Japan and IT Based Laboratory (ITBL) Project. Authors are grateful to Dr. Norihiro Nakajima, Dr. Yoshio Suzuki, Mr. Yukihiro Hasegawa and Mr. Nobuhiro Yamagishi.

7 References

Bakera, P. G., Brassa, A., Bechhoferb, S., Gobleb, C., Patonb, N. and Stevensb, R. (2000): Transparent Access to Multiple Bioinformatics Information Sources. *Bioinformatics,* **16**(2): 184-185.

Blythe, J., Deelman, E. and Gil, Y. (2004): Automatically Composed Workflows for Grid Environments. *IEEE Intelligent Systems,* **19**(4): 16-23.

Buetow, K. H. (2005): Cyberinfrastructure: Empowering a 'Third Way' in Biomedical Research. *SCIENCE,* **308**(5723): 821-824.

Cannataro, M., Comito, C., Schiavo, F. L. and Veltri, P. (2004): Proteus, a Grid Based Problem Solving Environment for Bioinformatics: Architecture and Experiments. *The IEEE Computational Intelligence Bulletin,* **3**(1): 7-18.

Foster, I., Kesselman, C. and Tuecke, S. (2001): The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing*, 1-4, Springer-Verlag.

Gil, Y., Deelman, E., Blythe, J., Kesselman, C. and Tangmunarunkit, H. (2004): Artificial Intelligence and Grids Workflow Planning and Beyond. *IEEE INTELLIGENT SYSTEMS*, **19**(1): 26-33.

Imamura, T., Yamagishi, N., Takemiya, H., Hasegawa, Y., Higuchi, K. and Nakajima, N. (2003): A Visual Resource Integration Environment for Distributed Applications on the Itbl System. *ISHPC 200*, Tokyo, Japan, 258-268.

Lacroix, Z. (2002): Biological Data Integration: Wrapping Data and Tools. *IEEE Trans Inf Technol Biomed,* **6**(2): 123-128.

Miles, S., Papay, J., Wroe, C., Lord, P., Goble, C. and Moreau, L. (2004). Semantic Description, Publication and Discovery of Workflows in Mygrid. *ECSTR-IAM04-001*, Electronics and Computer Science, University of Southampton.

Orchard, S., Montecchi-Palazzi, L., Hermjakob, H. and Apweiler, R. (2005): The Use of Common Ontologies and Controlled Vocabularies to Enable Data Exchange and Deposition for Complex Proteomic Experiments. *Pacific Symposium on Biocomputing*, Hawaii. USA.

Pytlinski, J., Skorwider, L., Bala, P., Nazaruk, M. and Wawruch, K. (2002): Biogrid-Uniform Platform for Biomolecular Applications. *Euro-Par200*, Paderborn, Germany, 881-884, Springer-Verlag GmbH.

Rowe, A., Kalaitzopoulos, D., Osmond, M., Ghanem, M. and Guo, Y. (2003): The Discovery Net System for High Throughput Bioinformatics. *Bioinformatics*, **19**(suppl.1): i225-i231.

Shah, S. P., Sawkins, D. Y. H. J. N., Druce, J. C., Quon, G., Lett, D., Zheng, G. X., Xu, T. and Ouellette, B. F. (2004): Pegasys: Software for Executing and Integrating Analyses of Biological Sequences. *BMC Bioinformatics*, **5**(40).

Stevens, R. D., Robinson, A. J. and Goble, C. A. (2003): Mygrid: Personalised Bioinformatics on the Information Grid. *Bioinformatics*, **19**(Suppl 1): i302-i304.

Sulistio, A., Poduvaly, G., Buyya, R. and Tham, C. K. (2005): Constructing a Grid Simulation with Differentiated Network Service Using Gridsim. *Proceedings of the 6th International Conference on Internet Computing (ICOMP'05)*, Las Vegas, USA.

Takemiya, H., Imamura, T. and Koide, H. (1999): Development of a Software System (Sta: Seamless Thinking Aid) for Distributed Parallel Scientific Computing. *IPSJ MAGAZINE*, **40**(11): 1104-1109.

Wroe, C., Goble, C., Greenwood, M., Lord, P., Miles, S., Papay, J., Payne, T. and Moreau, L. (2004): Automating Experiments Using Semantic Data on a Bioinformatics Grid. *IEEE INTELLIGENT SYSTEMS*, **19**(1): 48-55.