UNIVERSITY OF WASHINGTON

BOTHELL

# From Data to Discoveries:

# A White Paper

April 15, 2002

Professor Michael Stiber
Principal Investigator, Biocomputing Laboratory
Computing and Software Systems Program
18115 Campus Way NE, Box 358534
Bothell, WA 98011-8246 USA

http://depts.washington.edu/biocomp/

## Summary

Technological advances have enabled individuals and organizations to collect more and more data each day. While this data is recognized as a valuable asset, the challenge lies in leveraging this data into returns on assets. To do this, data must be turned into timely information that can influence decisions and enhance collaboration in a rapidly changing environment.

The process of transforming data into information often falls within the domain of investigators, specialists and experts. Investigators do the work of gathering data, experts' knowledge determines what types of operations and analyses should be done to the data, while specialists such as programmers write programs to automate operations and analyses. There is a mutual dependency:domain experts depend on programmers to automate operations and analyses they otherwise would have to do manually, programmers depend on domain expertise to create context for writing programs, and both depend on investigators to actually deliver data (who in turn require the others' help to plan and implement data analysis). Even though experts' knowledge and specialists' skills contribute to turning data into information, at the same time they form a bottleneck for turning data into information. Even if all three functions are embodied by one individual (or a small team), the time and effort required to "wear all three hats" — or, more likely, the sub-optimal performance of a generalist versus a specialist — still forms a bottleneck.

The bottleneck becomes more evident as organizations collect more and more data.The process of turning data into information is costly and time-consuming; the result can be irrelevant to decision-making needs. Many entities find themselves in a "data rich, information poor" state. The state represents an interesting opportunity for solutions that can mitigate the bottleneck.


## 1 The Challenge of Data Flux

Individuals and organizations are faced with a great dilemma, the result of a confluence of accelerating trends. These trends include:

- a rapid increase in the quantity, variety, and level of detail of data that can be gathered and stored,
- the continual development of new technology for processing this data,
- an expansion of knowledge unprecedented in human history.

Unfortunately, though these are parallel trends, their interaction has been rather limited. Our ability to convert *data* into *information* — by selecting, analyzing, and interpreting it — remains limited. It is very difficult for any individual to have both sufficiently broad and deep knowledge of the data and analysis methods to effectively use all this new-found "power". The typical approach that is taken is to arbitrarily simplify the data ("regularize" it) and choose a from a subset of analysis methods (favored tools; "best practices") for which one's knowledge is sufficient.

However, there may be a way to turn this dilemma into an opportunity. The same advances that drive this explosion of data, technique, and knowledge have also produced tools that can assist in intelligent data analysis and interpretation [1]. The goal of the LOGOS project is to develop a platform for experimentation, evaluation, and utilization of an integrated set of tools meant to manage the complete information life cycle, from collection, processing, analysis, visualization, inference, generalization, and dissemination of new results to review of previous results and the beginning of a new cycle.

## 2 From Data to Discovery

To set the stage for understanding this solution, Figure 1 establishes a simplified multidimensional space of data users. Individual users can be placed as a point within the three-dimensional space defined by: their degree of expertise within the domain of inquiry, their access to data, and their expertise *and access* to data analysis tools. While it is true that these dimensions are likely not independent of each other (for example, an expert in some field is likely to have access to data and analysis methods), it is common that they have a great deal of independence. It is also certainly true that time and effort spent along one of these dimensions is transferable to only a very small extent to the others (you may spend a lot of time learning about some problem area, but the learning process usually doesn't produce much in the way of real, usable data or tools).

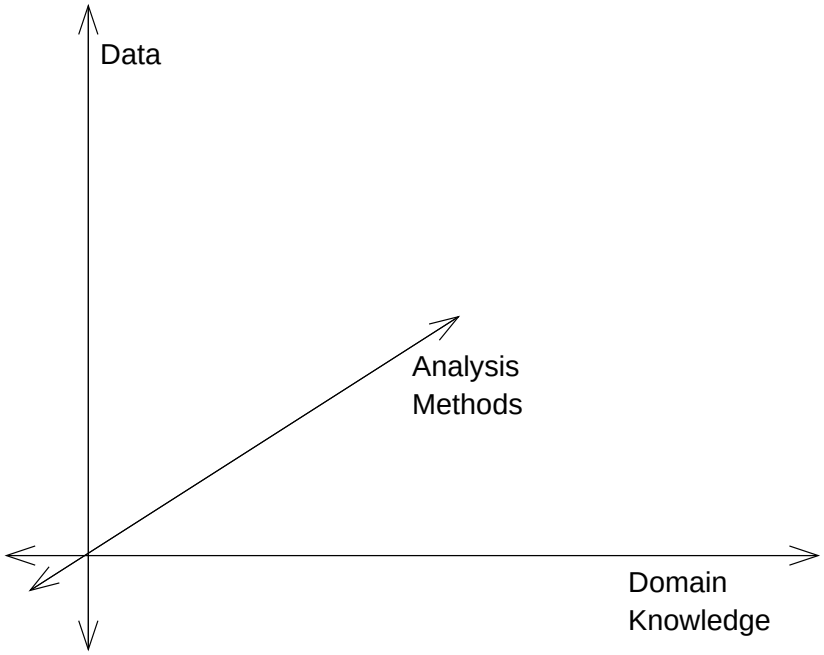


Figure 1: The multidimensional space of data users.

Along any of these dimensions, the number of people decreases as we move from "low" to "high". Domain experts must work with those with less expertise to assist in data acquisition and/or share data with other experts. Construction of large data libraries without specific motivation is probably not the most efficient use of scarce resources. The same might be said for unguided algorithm development. The typical situation is one in which an expert performs data acquisition, buys and customizes and/or develops analysis tools, and collaborates in an *ad hoc* manner with others to share data and tools. The result is one in which the "best practices", "best data", and "deepest knowledge" dwell in the community as a whole but are never brought together in one place.

To the resolve the bottleneck identified in the Summary, we envision a system which can greatly decouple these dimensions. We further envision a future where even fairly casual users — those at the "low end" of each scale — can obtain needed information. The LOGOS system will remove the sources of this bottleneck by:

1. Facilitating secure, on-the-fly collaboration through a distributed data repository. Users will be able to select from among their own data and data which others have made available.

2. Enabling users to choose operations needed for analyses without having to depend on others to write code each time they want to do analysis.

3. Providing users with a set of tools to answer specific questions by automating the construction of 'data to information' plans using knowledge captured from domain experts, and to perform exploratory analysis through interactive collaboration with the computer as an aid in formulating and evaluating various analysis approaches for turning data into information.

A "cartoon" session with LOGOS is presented in Figure 2, with details of system structure described in a later section. A user starts by wanting some information — he or she *asks a question* at some client machine (a). This question constrains the response display: it constrains the kind of answer that is appropriate. An agent is dispatched to locate domain knowledge relevant to this question on a distributed set of servers (b). The agent then carries this knowledge to a cycle server running a rule-based system, which, based on the question and knowledge (both domain knowledge gathered by the agent and user knowledge carried by the agent from the client) to produce a specification for the types of data and analysis tools required to produce candidates for possible responses (c). The agent is then re-dispatched to gather the needed data and analysis methods (d), carrying them to a (possibly another) cycle server to produce a set of responses to the original question which are presented to the user (e). These are then returned to the

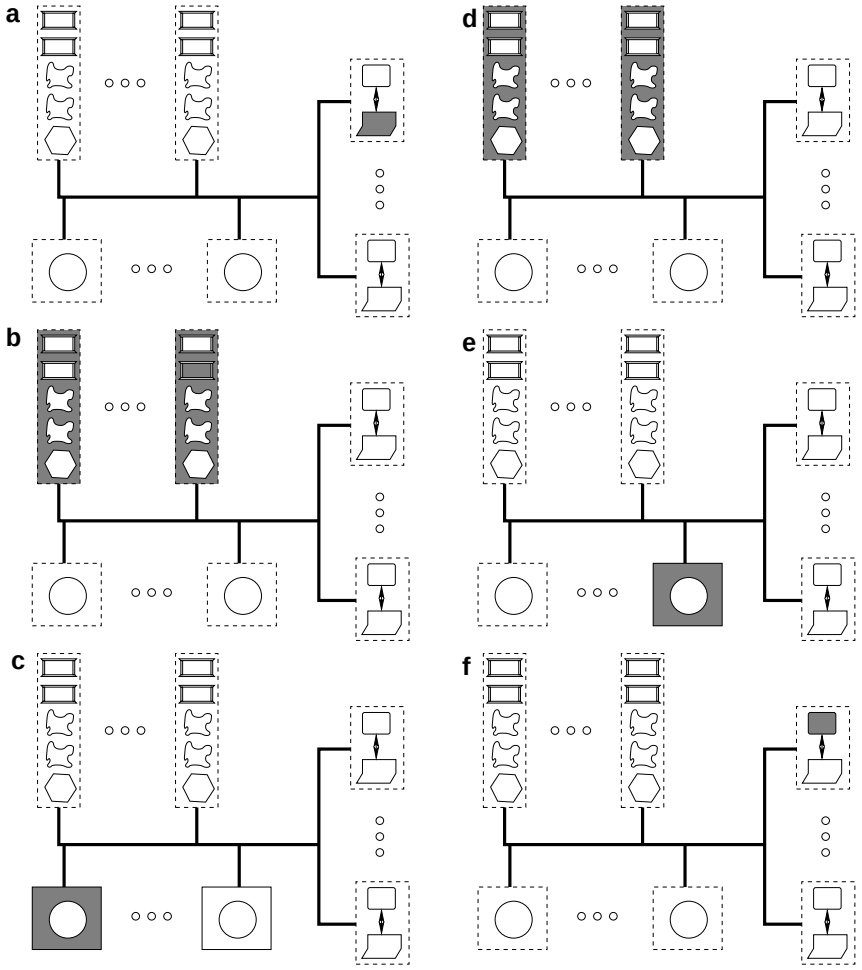


Figure 2: An example session.

user for further interaction, which could result in refinement of the answer, if necessary, leading to the final, desired information (f).

# 3 Example Application Areas

**Web Associate** HTML and browser software was originally based on the idea of *logical markup*. HTML was a markup language for indicating hypertext document structure, with physical appearance determined by the browser display software. As the WWW has developed, however, there has been a great push on the part of content providers towards visual markup capabilities, with the browser's task being to render a document as closely as possible to its specified appearance.

This evolution is a reasonable outgrowth of the desire to deliver a product with uniform quality and appearance. From the user's point of view, *if* that product is what one desires, then visual markup is an end-to-end solution. However, to the extent that the user wants *information*, formatted compactly and presented to maximize utility for particular tasks, visual markup is an obstacle. It is ironic that modern computer technology, often touted as enabling "mass customization" of physical products, is moving in the direction of "one size fits all" in the information arena.

A "Web Associate" is a networked computer system which works with a user to produce views of diverse, dynamic information tailored for specific tasks. Rather than just producing customized layouts as part of a pure information retrieval system, the Associate synthesizes multiple sources to deliver an "information view" that is task-specific (as opposed to information-source-specific). In terms of the LOGOS architecture, the source data is distributed throughout

the Internet (sometimes including metadata such as XML), abstraction methods are techniques for extracting, combining, and summarizing such data, domain knowledge includes rules relating to user tasks, and collaboration knowledge models the user's preferences and evaluations of various information sources. Since client desktop machines will quite likely be low-cost personal computers or thin-client appliances, the workspace sever allows computationally-intensive operations to be off-loaded.

**Scientist's Associate** Both the quantity of scientific data and the complexity of its analysis are almost unmatched in any other application. An example of this is the brain sciences [2, 3], in which the various anatomical, physiological, and functional components of nervous systems are studied to better understand how the low-level activity of individual cells maps to organisms' behaviors. Dataset sizes currently range up to multiple-petabyte levels. The data itself is diverse, including anatomies (2D and 3D images, 2D and 3D movies 3D geometries), physiology (time series, point processes), molecules (2D and 3D density distributions), and symbols (functions, behaviors). Scales range from Angstroms to meters and from microseconds to decades. Additionally, data is gathered from single cells in individual animals and can vary greatly from one animal to another and one experiment to another. However, researchers rarely want to ask questions about a particular cell or individual; they want to generalize from the examples they've seen to produce an understanding of how cells and systems function.

Thus, the situation researchers find themselves in is analogous to that of web users: an enormous flux of base data (which may vary in reliability according to the source, etc.) and a large gap between this data and the tasks to be performed. LOGOS addresses these needs at multiple levels:

- Source data can be locally collected in a laboratory and shared among multiple labs across the Internet. Eventually, this could include all published data, with "live" connections among the publications displays (graphs, images, tables), the analysis methods used to produce those displays, and the experimental data.
- Analysis methods are used for both detailed data analysis (including error tracking) and producing visual representations of collections of data.
- Domain knowledge represents declarative information, both general and specific, about the field of inquiry, experimental methods, analysis strategies, experimental design, workflow, etc.
- Collaboration knowledge models the user's view view of the field, including preferred techniques, data security, and data source evaluation.

Client machines used in research run the gamut from high-performance engineering workstations to commodity personal computers. A user's agent negotiates with workspace servers to flexibly allocate computational tasks according to the client's and the server's capabilities.

**Situation Awareness** *Situation awareness* addresses the general problem of summarizing (in a task-dependent manner) and communicating to humans an overview of the "environment" around them [4, 5]. Situation awareness allows humans to develop accurate *mental models* of their work environment. Extensive studies and systems development effort has been directed towards situation awareness and mental models in a variety of tasks, including aircraft piloting [6, 7], air traffic control [8, 9], nuclear power plant operation [10], and national defense [11, 12]. Less effort has been directed to other tasks, such as computer systems administration [13], mainly because of the perception that they are less obviously related to human safety rather than lack of applicability or need.

What all of these tasks have in common is that a human is interacting with a complex system/environment which includes a variety of agents (human, natural, and artificial) and from which large quantities of time-dependent information can be extracted. The challenge is to *fuse* this information into a unitary view of environment status that facilitates construction of useful mental models — models that allow both short-term, tactical and longer-term, strategic decision making. LOGOS can support situation awareness applications by:

- providing access to source data from a wide range of sensors,
- systematizing this data and applying domain knowledge to automate determination of data significance, correlations, etc.,
- "handicapping" data based on its source (the originating human or machine "collaborator"),

- and applying well-defined data abstraction rules to deliver information to the user in a useful form.

Clients in these systems will vary greatly, from small, portable, wireless devices worn by individuals all the way to complete, multiuser virtual reality systems.

# 4 The LOGOS System

The primary goals for LOGOS are:

- It should handle a wide (ideally, arbitrary) range of data, algorithms, and knowledge in a unified environment, and allow addition of data types as future data collection and computation require [14].
- It should include advanced tools for data collection.
- It should be able to access distributed data transparently.
- It should provide access control at the level of the individual data set to ensure data integrity and provide security to prevent public release of private or proprietary information [3].
- It should incorporate standard interfaces for user-developed algorithms, agents, simulators, etc [15].
- It should support multinational collaboration by separating language and data representation as much as technically feasible, segregating linguistic information into a separate resource manager within the user interface, and supporting future data types and knowledge representation techniques for further reduction in linguistic data dependencies.
- It should support heterogeneous hardware and software environments, including basic interoperation and data sharing with other tools via capability to transcode data among industry-standard data formats [16].
- It should provide control of the entire information life cycle, by integrating all post-data-collection operations, including analysis, knowledge-based inference, simulation, publication, and data collection design.
- It should preserve all base and derived information, regardless of its being superseded [15, 17] .
- It should allow high degree of user customization, not only of interface but also of how the system will interact with him/her [15].
- It should allow for the possibility of conflicting information, and provide user-configurable mechanisms for conflict resolution.
- It should track data accuracy from initial collection through all computations performed on it, providing direct support for probabilistic representations [3].
- It should utilize domain knowledge to automate inference operations.

The fundamental LOGOS architecture is designed to provide the flexibility and power needed to accomplish these goals. It is composed of the following major components:

**Data servers** It is envisioned that data and methods may be distributed among a number of organizations/laboratories/sensors. Each data server would be a repository for data (both base and derived), analysis methods, and knowledge. Knowledge includes knowledge about the data (or *metadata*), knowledge of the analysis methods, general knowledge about data analysis (strategies for developing analysis plans), and user-oriented knowledge, such as preferences, assessments of reliabilities of various data based on their sources, etc.

Each organization would provide a networked gateway device which would interact with agents to:

- authenticate the source of the agent,
- describe server content which the agent may access,
- and deliver information at agent request.

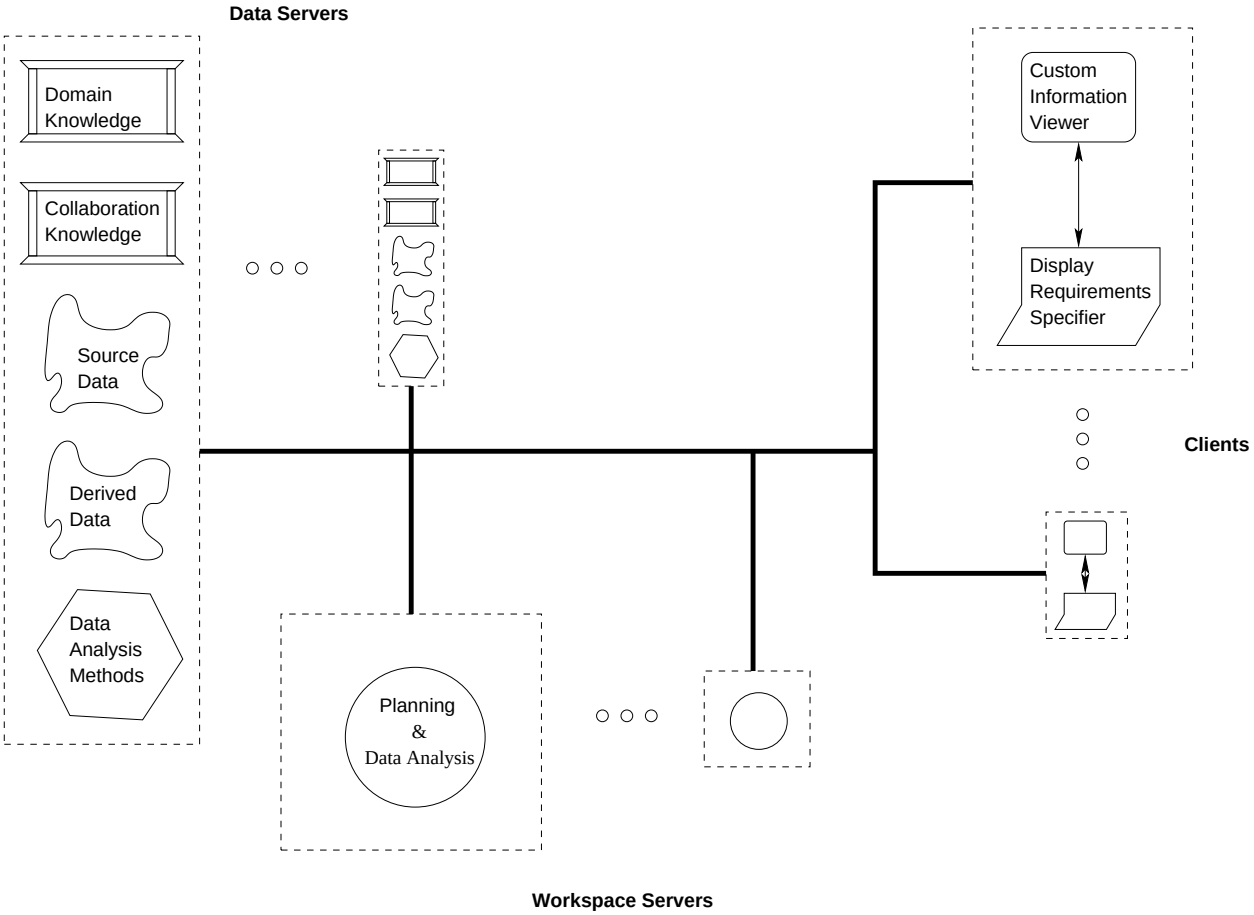


Figure 3: LOGOS Architecture.

**Workspace servers** These are machines made available for use as compute engines. A knowledge-based system interacts with agents to generate analysis plans (using knowledge obtained from the data servers); these plans are then implemented by the agent retrieving data and methods from the data servers, a server executing the analysis plan, and the agent presenting results to the clients (possibly storing derived data back into a data server).

**Clients** These are desktop machines which can interact with users to dispatch and receive agents to answer user questions. They provide the ability to (possibly collaboratively) specify desired result forms and display candidate and final results. Clients can also include interfaces to other systems, including simulators, etc.

# References

[1] D. J. Hand, "Intelligent data analysis: issues and opportunities," in *Advances in Intelligent Data Analysis: Reasoning about Data* (X. Liu, P. Cohen, and M. Berthold, eds.), pp. 1–14, Springer-Verlag, 1998.

[2] M. Huerta, S. Koslow, and A. Leshner, "The human brain project: an international resource," *Trends in Neurosciences*, vol. 16, no. 11, pp. 436–8, 1993.

[3] "Establishing identified neuron databases," workshop report, National Science Foundation, Arlington, VA, June 1994.

[4] N. Sarter and D. Woods, "Situation awareness: A critical but ill-defined phenomenon," *International Journal of Aviation Psychology*, vol. 1, no. 1, pp. 45–57, 1991.

[5] M. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, vol. 37, no. 1, pp. 85–114, 1996.

[6] N. Sarter and D. Woods, "Pilot interaction with cockpit automation: Operational experiences with the Flight Management System," *International Journal of Aviation Psychology*, vol. 2, no. 4, pp. 303–21, 1992.

[7] K. Mosier, L. Skitka, S. Heers, and M. Burdick, "Automation bias: Decision making and performance in high-tech cockpits," *International Journal of Aviation Psychology*, vol. 8, pp. 47–63, 1998.

[8] D. Hopkin, *Human Factors in Air Traffic Control*. London: Taylor and Francis, 1995.

[9] C. Wickens, A. Mavor, R. Parasuraman, and J. McGee, . *The future of air traffic control: Human operators and automation*. Washington, D.C.: National Academy Press, 1998.

[10] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human Factors*, vol. 39, pp. 230–53, 1997.

[11] K. T. Garner and T. J. Assenmacher, "Situational awareness guidelines," Tech. Rep. NAWCADPAX–96-268-TM, Naval Air Warfare Center Aircraft Division, Patuxent River, MD, Jan. 1997.

[12] Committee to Perform a Technology Assessment Focused on Logistics Support Requirements for Future Army Combat Systems, Board on Army Science and Technology, Commission on Engineering and Technical Systems, and National Research Council, *Reducing the Logistics Burden for the Army After Next: Doing More with Less*, ch. Situational Awareness, pp. 197–208. Washington, D.C.: The National Academy Press, 1999.

[13] D. Hrebec and M. Stiber, "A survey of system administrator mental models and situation awareness," in *Proc. SIGCPR 2001 Conference*, (San Diego, CA), ACM, Apr. submitted, 2001.

[14] N. Goodman, S. Rozen, and L. Stein, "Building a laboratory information system around a C++-based object-oriented DBMS," in *Proc. 20th VLDB Conf.*, (Santiago, Chile), pp. 722–9, 1994.

[15] J. C. French, A. K. Jones, and J. L. Pfaltz, "Scientific database management," Tech. Rep. 90-21, University of Virginia, Dept. of Computer Science, Aug. 1990.

[16] E. Mesrobian, R. Muntz, E. Shek, S. Nittel, and M. LaRouche, "OASIS: An open architecture scientific information system," in *Proc. RIDE '96*, (New Orleans), pp. 107–16, Feb. 1996.

[17] L. Kerschberg, H. Gomaa, D. Menasce, and J. Yoon, "Data and information architectures for large-scale distributed data intensive information systems," in *Proc. 8th Int. Conf. Scientific & Statistical Database Management*, (Stockholm), pp. 226–33, June 1996.