Prepared for:

National Institutes of Health
National Center for Research Resources

caBIG™: Opportunities and Challenges for Use Beyond Cancer

June 2006

The NIH National Center for Research Resources has contracted The MITRE Corporation to track developments and to inform the research community in the area of clinical research information technology through a series of targeted research reports.

Approved for Public Release; Distribution Unlimited, Document Number: 06-1520.

CENTER FOR ENTERPRISE MODERNIZATION

MITRE

Center for Enterprise Modernization
McLean, Virginia

National Institutes of Health National Center for Research Resources

The National Institutes of Health National Center for Research Resources (NCRR) provides laboratory scientists and clinical researchers with the environments and tools they need to understand, detect, treat, and prevent a wide range of diseases. With this support, scientists make biomedical discoveries, translate these findings to animal-based studies, and then apply them to patient-oriented research. Ultimately, these advances result in cures and treatments for both common and rare diseases. NCRR also connects researchers with one another and with patients and communities across the nation. These connections bring together innovative research teams and the power of shared resources, multiplying the opportunities to improve human health. For more information, visit www.ncrr.nih.gov.

Accelerating and Enhancing Research from Basic Discovery to Improved Patient Care

The MITRE Corporation

The MITRE Corporation is a private, independent, not-for-profit organization, chartered to work solely in the public interest. MITRE manages three federally funded research and development centers (FFRDC) and partners with government sponsors to support their critical operational missions and address issues of national importance. For more information about The MITRE Corporation and its work, visit www.mitre.org.

Applying Systems Engineering and Advanced Technology to Critical National Problems

The views expressed in written materials or publications do not necessarily reflect the official policies of the Department of Health and Human Services, nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

Table of Contents

1. Introduction 1

1.1 Key Concepts 1

2. caBIG Beyond Cancer 3

2.1 Biomedical Research as a National Institutes of Health–wide Objective 3

2.2 Non-Cancer Groups’ Participation in caBIG 3

2.2.1 National Heart, Lung, and Blood Institute 4

2.2.2 National Institute of Neurological Disorders and Stroke 4

2.2.3 Mayo Clinic 4

2.2.4 University of Pennsylvania 5

2.2.5 Georgetown University 5

2.2.6 Other Non-Cancer Activities 6

2.3 A Welcoming Environment for New Domains 6

2.3.1 Data Sharing Architecture and Principles 6

2.3.2 Extensibility of Tools 7

2.4 Expanding the System Architecture 8

2.4.1 Additional Data Format Definitions 8

3. Development Process for caBIG Expansion 10

3.1 Development Overview 10

3.2 Logical Modeling 10

3.2.1 Requirements Analysis 10

3.2.2 Use of Model Driven Architecture 11

3.2.3 Unified Modeling Language 11

3.3 Scaling the Vocabulary Service and Ontologies 13

3.4 Generating APIs and Software Coding 14

3.5 Testing and Evaluation 15

3.6 Monitoring the Health of the Grid: Capability and Service Availability 16

4. Technical Governance and Organizational Structures 18

4.1 Management of Open Source Development 18

4.2 Data Provenance 18

4.3 The Security and Privacy Model 19

5. Next Steps 21
5.1 Terminologies and Ontologies 21
5.2 Architecture and Development 21
5.3 Certification 23
6. Conclusion 24
Acronyms 25
Contributors 27
Endnotes 28

List of Tables

Table 4-1. Security and Privacy Requirements 19

1. Introduction

This is the second report on the Cancer Biomedical Informatics Grid™ (caBIG™) produced for the National Center for Research Resources (NCRR). The first report, *caBIG™ Overview*, which addresses the caBIG environment and tool sets, provides information on the program as a whole.[1] The report also describes the components of caBIG and governance of caBIG workspaces.

This document discusses the expansion of caBIG for use beyond the cancer domain and covers issues that are more technical in nature. Its purpose is to describe the technology capabilities and infrastructure required to expand caBIG beyond the cancer community to the entire biomedical community.

caBIG is being developed as a coordinated whole, not as a series of autonomous projects across independent institutions. This enables a unified approach, with unified governance among communities, and reduces the chances of mismatches in system components. Modules are written to a unified interface infrastructure, as described in the first report.

Teams of developers and adopters, working under contract, are developing caBIG. These teams, whose members initially came from academic medical centers only, are increasingly including developers from industry in response to open, competitive requests for proposals. The overall strategic plan for caBIG's expansion into the non-cancer domain might need to consider whether this model should be retained.

1.1 Key Concepts

To evaluate the technical structure of caBIG, some ideas need to be discussed up front.

caBIG incorporates several emerging technologies, such as the following:

- **Open source**, which is a method of designing, developing, and sharing computer software in a manner that is open to the public and available without charge. Open source systems generally evolve through community cooperation, and caBIG is modeled on this approach. Many major open source software products rely both on volunteer developers, who work on their own time, and paid contributors, who work on the software as a part of their company's strategic plan (e.g., IBM contributes a large amount of open source software to the Linux open source operating system because it fits into its overall technical strategy[2]). Open source systems generally publish their source code (the set of computer instructions that can be compiled and integrated into a working system) on the Internet under a licensing agreement that usually is without charge. Many open source communities provide extensive software training, documentation, and maintenance and enhancement capabilities. Examples of open source products are Linux (a major operating system), Mozilla (an Internet browser), and Mediawiki (which is used to run tools such as Wikipedia).
- **Service oriented architecture (SOA)**, which is a way of decomposing information technology (IT) applications into sets of interoperable services. For example, a Web site could provide a currency conversion service by linking to a service at a bank that provides conversions on demand. By using the bank's service, the Web site owner would not have to know anything about currency rates in today's market. Instead, the owner could take advantage of a published application programming interface (API) provided by the bank, submit the dollar amount to be converted and the currency of interest (e.g., the euro), and receive an answer, which is then provided to Web site users. Service offerers would

publish a set of metadata[3] in a registry that indicates the type of services they offer, where the services are located on the Web, how to call them, and the degree of accuracy or precision that can be expected from them. Systems such as GoogleEarth and Amazon.com offer many services over the Web. It is not necessary to use the Web to provide services, but many organizations do. Typically, a service encapsulates a business process, which may be very atomic (e.g., currency conversion) or more complex (e.g., arranging for payment via credit cards). caBIG services will be offered within a grid environment, as described in the next paragraph.

- **A grid**, which is a "hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."[4] Computer systems on the grid use a common set of tools, such as common security and authentication, to exchange services and data. caBIG participants can put data sets on the grid (called "exposing" the data), which then can be accessed via services, as described previously, by caBIG authorized users. Participants also can share access to computational resources (e.g., computer systems with available computing cycles) for performing large computations (e.g., protein folding calculations). The use of a grid essentially enables the creation of a virtual enterprise, in which all grid participants appear visible to one another but not to the outside world.

SOA and grids can be combined in the same enterprise. For example, the Starwood hotel chain is using a service-based grid to combine all of its legacy backend systems into a unified reservation system. This way, a customer can visit Starwood's Web site, request any available room in a city, and see available rooms from the other hotels Starwood owns (e.g., Westin, Sheraton). The grid is secure for the Starwood enterprise, and the services reach out to many legacy applications located in many different computer centers around the world in a manner that appears seamless to the customer.[5] SOA also can be used to create virtual enterprises without using a grid; such architectural tradeoffs can be performed as projects progress.

2. caBIG Beyond Cancer

Many caBIG activities, such as genomics research and clinical trials, could readily be expanded beyond the cancer domain. Several institutions either are already using caBIG for non-cancer applications or have been involved in caBIG to influence the development of tools appropriate for their non-cancer work.

caGrid, the technical infrastructure of caBIG, enables data sources to be exposed via services and facilitates queries. The infrastructure promotes the discovery and use of these sources to answer questions such as the following:

- How many data services from Cancer Center X are available?
- Which analytical services accept genes as inputs?
- Which services have metadata on macromolecules?[6]

Providing researchers with the ability to go beyond their own institutions to plug into a national grid to get answers to these types of questions could speed up the biomedical research process and even enable new research avenues to be developed. The existence of the Internet led to the development of new businesses, such as eBay and Amazon.com. Equivalently novel research paradigms could be created through the availability of new tools and federated resources.

The following sections discuss the rationale for extending caBIG beyond the cancer community and concomitant technical issues.

2.1 Biomedical Research as a National Institutes of Health–wide Objective

In March 2004, Elias Zerhouni, M.D., Director of the National Institutes of Health (NIH), described NIH's Roadmap for Clinical Research, "Re-Engineering the Clinical Research Enterprise." A major premise of the roadmap is the need for translational research, in which "bench" research is matured to provide clinical, or "bedside," results and, equally important, returns the knowledge and information gained from the clinical applications to the basic research community in an actionable manner. NIH, through the NCRR, created institutional Clinical and Translational Science Awards (CTSA) to respond to this need. CTSAs provide consolidated resources to academic centers, from clinical to scientific to computational, to enable integrated translational research.[7] In addition, many NIH institutes have changed their grant requirements to prioritize projects that contribute to translational research.

2.2 Non-Cancer Groups' Participation in caBIG

Several non-cancer groups are participating in caBIG as developers, adopters, or members of working groups. Some are using the Cancer Data Standards Repository (caDSR) and Enterprise Vocabulary Services (EVS) of caBIG. Typically, non-cancer adopters are using caBIG because it is easier—rather than starting from scratch—to implement one tool set across their institutions uniformly (regardless of whether users support cancer research or other programs) or because the caBIG infrastructure and tool set provide good support for their current work. The open source nature of caBIG makes this sort of tool set dissemination possible because there is no acquisition cost to the institution to download, modify, and test the software and because the institution can select the components that fit its particular needs (in other words, the Cancer Tissue Database [caTISSUE] component of caBIG can be leveraged without using the imaging components and vice versa).

2.2.1 National Heart, Lung, and Blood Institute

Recent efforts at the National Heart, Lung, and Blood Institute (NHLBI) either (1) have required that grantees remain abreast of caBIG developments (e.g., the Large-Scale Genotyping Center) in order to leverage them or (2) have specified that databases under development be created using the data definitions in the caBIG model. NHLBI also participated in caDSR development by submitting a specific corpus of information in a domain-specific work area.[8] This is a limited participation level, but it could be greatly expanded as caBIG expands beyond the cancer domain.

2.2.2 National Institute of Neurological Disorders and Stroke

Researchers at the National Institute of Neurological Disorders and Stroke (NINDS) have collaborated on multiple parts of the caBIG package. NINDS was a major contributor to caIntegrator, a tool developed to enable this type of collaboration.

To facilitate research into the relationship between primary brain tumors and either cancer or neurological disorders, the National Cancer Institute (NCI) and NINDS developed the Repository for Molecular Brain Neoplasia Data (REMBRANDT) as part of the caBIG tool suite.[9] The goal of this initiative is to create a publicly available database to house biological and clinical data on primary brain tumors. NINDS researchers also have made their presence felt in the caDSR community by helping define and relate terminology at the intersection of cancer and neurological diseases.

2.2.3 Mayo Clinic

The Mayo Clinic is a major caBIG stakeholder. The clinic is heavily involved in seven of the nine elements of caBIG:

- Architecture
- Data Sharing and Intellectual Capital
- Strategic Planning
- Tissue Banks and Pathology Tools
- Training
- Vocabularies and Common Data Elements (VCDE)
- Integrative Cancer Research

The clinic's involvement in so many elements of the program has exposed the entire Mayo research community, both cancer and non-cancer, to these elements. The clinic's approach to tool development is to be as non-disease-specific as possible so that the entire Mayo research and clinical community can benefit by leveraging caBIG tools.

LexBIG,[10] a tool developed by the Mayo Clinic (and supported in part by the caBIG program), is an outcome of this approach. LexBIG enables the visualization and manipulation of multiple terminologies and ontologies through a common API. Non-cancer terminologies, such as radLEX, a radiology lexicon, have been brought into EVS through this effort. The creation of this tool also facilitated the inclusion of other communities, such as the National Library of Medicine's National Center for Biomedical Ontologies, in the caBIG community.

Throughout the Mayo Clinic, caBIG tools are being adapted and adopted for non-cancer research, including cardiovascular and neuroscientific (e.g., Parkinson's disease, Alzheimer's disease) research and clinical trial support. As these communities continue to get involved in caBIG, their feedback will broaden its applicability.[11]

2.2.4 University of Pennsylvania

The University of Pennsylvania (Penn) also is deeply involved in caBIG. The university maintains an active presence in the following caBIG elements:

- Clinical Trial Management Systems
- Data Sharing and Intellectual Capital
- Imaging
- Integrative Cancer Research
- Strategic Planning
- Tissue Banks and Pathology Tools
- Training
- Architecture

From the beginning of the caBIG program, Penn has anticipated adopting caBIG tools across its enterprise and has been using cancer research as a test bed.

Penn plays a large part in the development and maintenance of caTISSUE, which is used to manage tissue banks. Penn's repository of more than 600,000 paraffin tissue blocks, dating back to the 1940s, has provided an excellent data set on which to develop and test tissue management applications. The tissues, from patients with various diseases, test the capabilities of caTISSUE to respond to both cancer research and non-cancer research management issues. Penn is considering using caTISSUE to support the tissue banking requirements for its entire medical center.[12]

2.2.5 Georgetown University

Georgetown University is involved in three caBIG elements:

- Architecture
- Clinical Trial Management Systems
- VCDE

Most of the tools developed at Georgetown University, either by the university or in collaboration with other institutions, are non-cancer specific. For example, Georgetown researchers helped design and implement the Visual and Statistical Data Analyzer (VISDA), an analytical tool for experimental data cluster modeling, visualization, and discovery. The tool is used for cancer and non-cancer research applications.

Georgetown University personnel also were crucial in the development of four new points of integration for GoMiner,[13] a non-cancer-specific tool that uses the Gene Ontology[14] to establish lists of functionally related genes derived from large experimental genomic data sets, regardless of condition or disease.[15]

2.2.6 Other Non-Cancer Activities

Other groups outside the cancer community are participating in caBIG. For example, the following active work was reported at a recent NIH Roadmap meeting:[16]

- Cardiology data standards work with the NHLBI, the Society of Thoracic Surgeons, the CRUSADE registry, the European Society of Cardiology, GlaxoSmithKline, and Duke University is being integrated into caDSR via the EVS.[17]
- Pennsylvania State University is using the Adverse Event Reporting System (AERS) module for reporting results of urologic pelvic pain clinical trials and developing a pilot project for testosterone and growth hormone studies.[18]
- The University of Minnesota is using Unified Modeling Language (UML) and caBIG to create a standardized terminology that can be used for primary-care-based clinical trials.[19,20]

2.3 A Welcoming Environment for New Domains

To extend into new domains, caBIG needs to expand in every aspect—from architecture to vocabularies. The sections that follow describe some of the key technical considerations for accomplishing that expansion.

2.3.1 Data Sharing Architecture and Principles

Using a federated approach to data storage and access was a key decision in the caBIG design. This approach enables each institution (e.g., a genomics laboratory) to generate its data and update the databases that its researchers select for exposure on the grid. The data remains on the institution's own hardware, under the institution's control and in accordance with local policies and institutional review board guidelines. The institution's researchers use caBIG tools to harmonize their data with the caBIG standard terminology so that the data are compatible grid-wide for use by the entire research community.

Another approach—a centralized repository (or mediated) approach—could have been taken. A centralized repository would have required that data, whose access is controlled by a central maintenance administrator, be stored in a centralized caBIG repository. This approach is used to manage data within many enterprises. It also is used in some highly successful regional health information organizations to enable the sharing of electronic health record data among states. The benefit of the centralized repository approach is that groups of researchers can develop efficient tools, filters, and infrastructure to enable them to read all incoming messages and store them in a standardized repository, which can be optimized for efficient query responses. Query results can be analyzed and returned to the requestor in integrated form. Both approaches have positive and negative aspects, and the choice of one versus the other is dictated by priorities, costs, response time requirements, and security and data ownership considerations.

The decision by caBIG leaders to use open source format promoted widespread caBIG adoption. Models, applications, and terminologies are available in open source format so that organizations can adopt the tools at their own pace, in a sequence that makes sense to them.

The open source nature of caBIG enables institutions to download caBIG components, compile them, and use them inside their own firewalls, without exposing their applications and associated data on the grid. This selective use of caBIG components could be useful for the biomedical research community because it enables institutions that cannot otherwise afford biomedical research tools to obtain them. Users do not have to be on the grid to adopt open source tools for their internal enterprise systems; therefore, caBIG tool adopters in the biomedical research community do not have to expose their data for use by other organizations on the grid.

2.3.2 Extensibility of Tools

Many tools in the current caBIG environment, such as support for proteomics data analysis, are written in a generalized manner that only focuses on cancer specifics where absolutely necessary. The tools are applicable to the analysis of certain experimental data types, regardless of the disease being researched. This flexibility enables the tools to be easily moved into other domains. For example, caMassClass,[21] which was developed by the caBIG community, is one of the first comprehensive statistical analysis packages contributed to R[22] for processing and classifying protein mass spectra (e.g., Surface-enhanced Laser Desorption Ionization [SELDI]) data). This library of functions is generalized (beyond cancer-specific proteomics research) to support typical analytical pipeline procedures, such as processing high-dimensional files, aligning peaks, normalizing, adjusting mass drift, selecting features, classifying, and outputting various file types.

As the biomedical research community begins to engage with caBIG, new toolkits must be developed, or current toolkits must be extended, to cover expanded imaging capabilities, genomics, proteomics, metabolomics, and so on. It may be possible to accelerate the utility of the grid by surveying current tools being used in non-cancer research laboratories, wrapping them in caBIG-compatible interfaces, and enabling them to be used on the grid. This could make caBIG more attractive to other research communities because new users would find on the grid the tools they are already accustomed to using in their labs.

The initial development of caBIG was based primarily on academic institutions. Industrial partners such as tool vendors are increasingly becoming involved. This is beneficial for the following reasons:

- Adding new contributors can provide a critical mass of data sharing that creates a new market in the private sector, in much the same way the Internet created a critical mass of data sharing that enabled the creation of businesses that previously would have been inconceivable (e.g., eBay). Vendors of tools and devices can expand their markets or create novel products that are not currently feasible, thus furthering research.
- The use of a standardized development and data environment can enable vendors to redirect developers from creating point-to-point interfaces to creating specific tools for which generalized, standards-based interfaces are used to expose the tools on the grid. caBIG provides an infrastructure for that type of interface and encourages vendors to use it. A mechanism exists for exposing data for interface purposes while retaining commercial rights and licensing, which provides a continuing incentive for innovation.

The potentially large scale of an expanded caBIG environment may offer other benefits, such as offsetting the costs of long-term support. When a new software component is created within a small community for its exclusive use, the cost of long-term support must be considered. If the component is never used outside the small community that created it, these costs, which can grow substantially for large suites of components, must be borne entirely by that community. This problem is aggravated if the community consists primarily of users who do not view software development and support as their primary tasks and thus may have difficulty providing the continuity of knowledge needed to support the software after it is created.

One of the fundamental reasons for the existence of successful open source software products is that the open source model drastically reduces the cost of long-term support by distributing such costs across a global community of users with similar needs, thereby averaging the costs over a much larger number of users. In addition, the knowledge needed to support the software is moved into the community itself, thus ensuring its preservation. Expanding caBIG beyond the cancer domain, therefore, can build the community of users needed for successful tool enhancement and maintenance.

2.4 Expanding the System Architecture

Rapid response time is one of the most commonly requested features in any computer system. Maintaining acceptable performance as the caBIG workload expands will be critical to maintaining the current user base while attracting new users. Several issues must be considered when expanding the grid.

The expansion of caBIG across the entire biomedical enterprise could enable the development of a centralized security and terminology infrastructure, which would free developers of individual components from having to reinvent the wheel. Of course, it is critical to assure the biomedical community that the grid is trustworthy and reliable enough to promote confidence in the integrity of the data during transmission and retrieval. A performance and reliability analysis for the entire system should be undertaken to investigate issues such as network stability and robustness against cascading failures.

As additional data sources begin to populate the grid, service discovery capabilities should increase accordingly, to support additional metadata requirements and enhanced query performance. This enhancement will support access to multiple data stores by a single query without requiring the user to know the underlying implementation details. In addition, computationally intense queries will be able to be shared across multiple machines using workflow techniques such as Business Process Execution Language (BPEL).[23] A sound approach for creating a large federation of data across a grid must be developed because this rarely has been attempted on the scale needed to accommodate the entire biomedical community.[24] Federated queries have been performed successfully across a prototype of the current grid, but tests should continue to be performed as the grid grows.

The transition of legacy systems to comply with caBIG requirements has begun by wrapping applications with APIs so that the interfaces to the legacy systems are available in caBIG-compliant forms. Not all legacy systems, however, can interact with such APIs; therefore, other techniques may have to be used or new tools may have to be developed to replace the legacy systems. Another concern is that it may be difficult to achieve the desired performance from wrapped legacy systems, especially when users perform federated queries across multiple data sources.

2.4.1 Additional Data Format Definitions

Developing common formats for representing experimental data is a difficult and iterative process. Standards for certain data types have been developed and are generally accepted, such as the Minimum

Information About a Microarray Experiment (MIAME) standard for microarray data. However, standards for most data types have not been developed and accepted. The wider user community must be engaged in the development of new standards to ensure that as many points of view as possible are represented. As caBIG expands to embrace the needs of the biomedical community, standard formats will need to be developed and adopted for a multitude of data types, including electroencephalogram (EEG) data, computational models of cellular processes, magnetic resonance imaging (MRI) images, and video.

Based on the amount of time and effort it took to develop existing standards, developing standards for all data types used in biomedical research promises to be a difficult task that will require input from a variety of sources—and compromise on all sides. A biomedical community organization that can identify necessary standards (or areas of overlap) and convene appropriate standards development organizations (SDO) and specialty societies to fill in the gaps would be invaluable. Without the organizing body, the SDOs will not prioritize the standards needed to complete the biomedical research instrumentation. Because the SDOs have limited resources, it is essential that they know where the greatest need exists.

3. Development Process for caBIG Expansion

The success of the expanded caBIG depends on effective approaches to ensure the usability, performance, security, and cost effectiveness of caBIG components. As caBIG continues to grow, it will be necessary to establish governance from a system-of-systems perspective—facilitating interoperability while continuing to support the managerial independence of local communities. These concepts are discussed in this section in terms of what is needed for caBIG to expand beyond the cancer domain.

The development process and tools also may need to be updated as the caBIG program evolves. Technologies for the SOAs and the Semantic Web are rapidly evolving. In addition, new tools are becoming available for software development and for managing security and privacy. As part of the governance process, it is important to periodically assess the technical approach to ensure that it remains relevant—not so far ahead that it uses immature technology, but not so far behind that it fails to take advantage of relevant work.

3.1 Development Overview

caBIG's development process is as follows:

1. A developer uses a methodology to construct a logical model, generally working with a community of other developers working on (or interested in) systems using similar data models.
2. An API is written to permit the component to be accessed via a service call. This process is aided through the use of the caCORE Software Development Kit, which enables the automatic generation of API code based on the logical model.
3. The logical model is entered into the metadata registry.
4. The logical model is annotated with controlled terminology to enable the interpretation of inputs and outputs.

Much of this process is manual, and the existing tools will need to be generalized to apply to a broader set of applications. Similarly, the code generator that creates code directly from the model artifacts to ease production and maintenance may have to be updated with additional information from the biomedical community.

3.2 Logical Modeling

3.2.1 Requirements Analysis

Before a new caBIG component is developed, an extensive user requirements process is undertaken, using modeling tools. Some caBIG users have described the process of requirements gathering, including gathering input from across the biomedical community, as extremely rigorous. Other caBIG users have said that the process must be substantially expanded because of the inherent workflow differences in user communities. As caBIG expands beyond the cancer community, creating an effective user requirements process will require input from other disease-related communities and organizational entities (e.g., industry, academia, government).

Current caBIG-funded developers are housed within research institutions, where they have ready access to prospective users. This does not mean that the process of translating many user requirements into a form that is useful to software developers is easy, and the fact that multiple communities, using multiple processes, will have to be accommodated to expand caBIG beyond the cancer community is daunting. Establishing clear lines of communication that enable developers to understand the working environments and data needs of end users could be a critical success factor. Some technologists suggest that an incremental approach be taken, in which simple requirements are implemented first, then enhanced over time as users are able to test the processes and tools and provide feedback about caBIG features, functions, user interfaces, and utility.

The caBIG program is implementing a set of development standards around the Unified Process Framework, which will increasingly drive projects supported by caBIG to use a standards-based incremental approach.

3.2.2 Use of Model Driven Architecture

The goal of caGrid is to seamlessly integrate computation, data storage, and analysis services across multiple nodes, often located at different institutions. To facilitate interoperability among participants, caBIG is using the Model Driven Architecture (MDA) paradigm, which permits business and application logic to be separated from specific implementation technologies.[25] Using MDA, and working in a manner compliant with modeling standards such as UML, various tools and technologies are used to create software.

The Cancer Bioinformatics Infrastructure Objects (caBIO) model and architecture is a good example of the use of the MDA approach within caBIG. caBIO provides standard object models and a uniform programmatic interface that provides access to caCORE technologies. (caCORE is a software development toolkit that supports the development of caBIG tools.) caBIO also provides an abstraction layer that enables developers to access genomic, systems biology, clinical and pre-clinical, and biomedical metadata, as well as a wide variety of biomedical terminologies.[26]

Analysis for developing the models begins with use cases to identify actors, relationships, and courses of action to follow when accessing caBIO resources. Details of the use cases are presented in UML diagrams, such as class, activity, and sequence diagrams. Software generation tools can then generate programs directly from these UML-compliant models in a manner that facilitates ease of both production and maintenance. Using this technique, caBIO creates APIs for a number of communication protocols and computer languages.[27]

3.2.3 Unified Modeling Language

The caBIG community has embraced UML as the standard language for specifying, visualizing, constructing, and documenting the artifacts of its software systems.[28] The language enables developers to use component metadata to learn how to use a service as well as to modify or extend the service. UML is one of the few common languages able to provide these services, and it has performed well in the current caBIG architecture. The use of this structured approach enables the design to be visible to the community and enables the design to be analyzed and improved.

Volunteer developers in the open source community, however, often do not use UML or any other modeling tools to document their work. Instead, they tend to include their documentation within the source code files or as text in documents. Once significant numbers of volunteer developers participate in caBIG, it may be necessary to work with them to find ways to encourage them to develop and maintain the UML models.

Although UML notations are powerful, few organizations have processes for using them that are mature enough to ensure that the organizations do not end up wasting resources in non-productive modeling activities. The effective use of UML results in two main classes of artifacts:

- **Needs capture**, which is the precise capture of needs that otherwise would have been missed or misunderstood
- **Design metadata capture**, which is the precise capture of design metadata for which no effective automated mechanism of capture currently exists

A developer should be able to use design metadata to understand more quickly how to modify or extend the software without damaging it. It is important to keep in mind, however, that using UML to capture design metadata is preferable only when there is no effective automated process for capturing the metadata. The automated capture of design metadata is the better choice, when it is available, because an automated (generator) process that captures metadata is not subject to the errors and misinterpretations that occur when humans process metadata. Automated capture also is far less expensive, easier to replicate on new systems, and orders of magnitude faster.

When UML is used ineffectively, modeling activities tend to be non-productive. The results of ineffective UML needs and design metadata capture are as follows:

- **Needs-noise amplification**, which occurs when UML captures problem details that are inaccurate, exaggerated in importance, or over-specified to the point that their implementation would result in extremely fragile, poorly generalized software. The term “amplification” is used because the act of capturing such information in a formal model unavoidably exaggerates the information’s importance and makes it more difficult to recognize the information as noise during the rest of the development process.
- **Design-noise amplification**, which occurs when UML captures design “guesses” or other forms of elaboration that compete with coding instead of generalizing it. A common form of this problem occurs when designers who are unfamiliar with a target system assume that they can safely elaborate a design to the point that the coding effort should be trivial, then hand the design over to a programming team that is far more familiar with the weaknesses, intricacies, relative strengths, resources, and variability over time of the target system and software environment. Most of these over-elaborated designs end up being discarded or severely modified to make them work, often at the cost of the coding team spending more time and putting forth more effort than would have been required with a non-noisy UML design.

An immature UML-using process that allows too much needs noise and design noise to enter is similar to a car whose driver randomly drives in the general direction of the destination instead of following a carefully laid out, optimal path. Such a process can be extremely slow and costly, and it is likely to fail because most resources are sapped by the resulting unproductive work. Even worse, the noise within a UML process can interact with itself to create additional levels of noise. This could result, for example, in the development of entire software modules to address needs that never existed in the original process but that seemed like reasonable inferences from earlier UML models. In a worst-case scenario, the result is a runaway process in which UML development never really ends and coding never really begins, all in the name of ensuring accuracy in meeting requirements that are, in effect, a fiction generated by the UML process itself.

Such issues emphasize the importance of providing a well-led group of architects and methodologists to support the overall caBIG development infrastructure by defining the best tools and how to use them, monitoring contributions, and testing and enhancing the development process as it emerges. If the methodology is allowed to be haphazard, the grid and its associated tools will become unstable and unreliable.

3.3 Scaling the Vocabulary Service and Ontologies

To promote data interoperability, caBIG projects register their data models as common data elements (CDE) in the caDSR. Common semantics are enforced by linking data models to underlying semantic concepts registered in the EVS. This means that a concept such as "neoplasm" will be defined in all caBIG systems in a standard way and that all systems exposed on the grid will use the same definition.

For biomedical use, the caDSR needs to be expanded to include concepts and data definitions that go beyond the cancer domain. This will be no small task. Thousands of data elements will have to be included, and their definitions will have to be standardized. Redundant concepts will need clarification, and other concepts will not fit neatly into existing taxonomies.

The challenge of managing the data structures and ontologies should not be underestimated. The work cannot be avoided; however, new tools are being developed that can assist. But either a real commitment must be obtained for developing a shared view of the data across the entire biomedical community or a technical means must be developed for traversing non-standardized domains. Both these tasks can be accomplished, but neither is simple or inexpensive. Managing data structures and ontologies will be a key issue in the success of caBIG in a biomedical environment.

To address the myriad ways in which similar concepts can be expressed, the NCI community developed the EVS for caBIG, which includes the NCI Thesaurus and the NCI Metathesaurus. The NCI Thesaurus was initially populated with concepts and terms from NCI departments and divisions, but it has evolved as additional researchers have interacted with the vocabulary. These interactions resulted in the addition of new concepts and terms and more refined relationships among existing concepts. The NCI Metathesaurus contains all public domain vocabularies from the National Library of Medicine Unified Medical Language System (UMLS) Metathesaurus[29] as well as NCI-specific vocabularies. It also contains additional proprietary vocabularies, such as the International Classification of Diseases (ICD-10 and ICD-O-3) and the Medical Dictionary for Regulatory Activities (MedDRA).[30]

caGrid participants describe their data resources as a collection of CDEs. Each CDE provides the element's conceptual definition and a description of acceptable values. These CDE components are linked to a controlled terminology, which is accessible using the EVS. Adopting a common terminology permits other caBIG participants to search for resources relevant to their needs in a more effective manner.

Establishing linkages between CDEs and the NCI Thesaurus requires several iterations between NCI curators and CDE developers. Bottlenecks are possible when there are many new candidate CDEs and/or inexperienced CDE developers. Each iteration involves a manual inspection of the proposed linkages for correctness. If appropriate concepts or terms are not available in the EVS, curators must conduct the research necessary to add them.

Expanding the EVS to incorporate the needs of the biomedical community presents several issues, primarily related to the availability of trained human resources and tools:

- Expanding the focus of EVS beyond cancer will require additional curators. The National Center for Bioinformatics employs approximately 15 curators to maintain the NCI Thesaurus. The curators are experienced subject matter experts who are trained in terminology curation. The majority have advanced degrees and practical scientific or clinical experience. It takes, on average, between 2 and 3 years for most inexperienced curators to develop into independent practitioners. Therefore, an expanded caBIG using the current EVS development paradigm will require many trained curators, even allowing for overlap in concepts from one domain to the next. The curators will need to confirm that concepts that are considered to be the same across domains are, in fact, the same across domains.
- Terminologies and ontologies that use standard languages such as Web Ontology Language (OWL) or are capable of being exported to the Resource Description Framework (RDF)[31] need to be developed. The NCI Thesaurus is maintained with a proprietary description logic tool (Apelon Terminology Development Environment) and housed in a proprietary database (Oracle). The NCI Thesaurus is available in an OWL Lite[32] format that is viewable and modifiable in the open source tool Protégé.[33] Converting the NCI Thesaurus from a proprietary version of Extensible Markup Language (XML) to OWL Lite was an admirable, leading-edge project. The effort presented challenges that exemplified the need to develop the aforementioned terminologies and ontologies. LexBIG,[34] a tool developed by the Mayo Clinic (and supported in part by the caBIG program) that enables the visualization and manipulation of multiple terminologies and ontologies through a common API, may become an important part of making the technologies and information used by caBIG available to the larger biomedical community.
- As additional communities become involved in caBIG, they will need to establish their own domain-specific ontologies.[35] A decision will need to be made whether to merge independently developed ontologies into a single large ontology, to couple them loosely using concept mappings, or to centralize terminology services. The importance of this architectural decision and the need for appropriate tools to implement and manage the ontologies cannot be overstated. Each domain specialty has its own view of data. For example, the structure and content of information required to describe a tissue sample to a pathologist is different from that used by a genomicist or an oncologist, yet must be unified in some way so that each specialist can find the information he or she needs about a specific sample in caTISSUE.

3.4 Generating APIs and Software Coding

The CDEs and UML models feed a software development toolkit that uses a software generator to provide developers with a standardized data structure and caBIG-compliant API. Developers then use the aforementioned UML models as guidance as they write the code needed to use the data structures. Developers may write code to accomplish the following:

- Link existing systems or tools to the grid by populating the data structure and creating a service to interface with the data.
- Develop new capabilities (new functions or services) that create data that can be exposed on the grid or used as a service. These capabilities typically are written in Java, a language that can be used on many different operating systems and computer platforms.

Developers draw down existing software they need from the caBIG gForge Web site. When they finish their new software or their changes to existing software, they place their work in the controlled repository, which in turn is used to create the distributions for software testing and delivery.

Many well-regarded open source development groups not only have expert developers inspect contributed software's source code frequently as part of the code acceptance process, but also make such inspections the core of the design process. The groups focus on regular inspection because most open source efforts rely heavily on software-attached documentation or, for some languages and efforts, the semantics of the programming language that was used. Because of this focus on internal documentation, open source software tends to evolve over time in a way that makes it increasingly accessible to new users and thus easier to review.

3.5 Testing and Evaluation

The biomedical community will have to consider a number of unique challenges associated with the testing of distributed systems before the systems can be deployed:[36]

- **Testing each component.** Does this node (database, consumer console, service provider) perform its function properly (as expected and according to specifications)?
- **Testing services and transport components that are working together as different subnetworks.** Do these services and components work in an integrated manner on the network as expected?
- **Testing each system's use of the network.** Does the network architecture have bottlenecks? What is the maximum volume it can handle? How is the network performing?
- **Testing the end-to-end suite of systems over the network.** Do these systems integrate as expected? Is the response time acceptable? Is the security infrastructure working as expected?

These challenges are addressed well in the current caBIG environment through several stages of testing. The stages range from unit testing of individual components to integration testing, system testing, and production testing. Multiple test cases are created for each use case, exercised by test scripts deployed to each test server, and completed with all MDA artifacts under configuration management control. Recent testing has focused on the performance of individual software applications. As caBIG evolves, the additional testing measures outlined previously may need to be considered to ensure a successful product across the entire biomedical community. The test suites also will have to be expanded to cover the vocabularies and domain constraints required for non-cancer applications.

caBIG has developed a clever strategy for maintaining a low incidence of software errors, which is critical to maintaining credibility in the biomedical community. The use of standardized tools has maximized the reuse of existing code, which has been tested and integrated into the existing system. An emphasis has been placed on developing various test beds and testing procedures. The caBIG strategy of contracting with developers, who create the software, and with adopters, who are charged with using, testing, and helping improve the software, has provided a realistic test environment. It has been suggested that a second tier of adopters be created, one in which new users, who are more representative of end users in terms of their IT expertise, are employed to further test the usability of the components. It also has been suggested that a systems integrator be provided, as is done with many large-scale systems, to ensure that all the various components work together as planned. In other words, a systems integrator would ensure that the design constraints of the overall architecture are followed and that components developed by disparate development groups do not diverge. The integrator also could maintain a unified central test bed that could serve as the baseline for all developers to test against.

3.6 Monitoring the Health of the Grid: Capability and Service Availability

The success of caBIG in a biomedical environment depends on many factors, but the performance of the grid is among the most important. Performance commitments are defined in system architectures, which describe how well the network is expected to respond to various requests on the system, such as a request for a data set or for access to a computational resource.

Several aspects of performance need to be considered. For instance, performance is tightly tied to the design of the architecture, but it also depends on various component developers following caBIG design criteria when developing their applications. There is no current enforcement of design principles in caBIG (as is done in commercial software development and in some open source communities) other than certifying the interoperability of data elements. Poorly designed and implemented code can completely clog a network. Biomedical applications will certainly add more load to the caGrid infrastructure; therefore, a method of governance that ensures that architecture constraints are followed may be useful. (See the previous discussion on the potential role of a systems integrator.)

The availability of both data and analysis services, and the ability of the user to discover which data and services are available, also is important. Metrics could be established to measure the availability of the services and data provided to ensure that those that are advertised are actually available. Some form of network responsibility should be established so that problems can be identified, tracked, and resolved.

It would be useful to review how services are used within the grid to discuss the performance monitoring environment. As a new service is connected, the owner of the service would be expected to provide an advertisement in the registry that is based on the Common Service Metadata standard associated with all caBIG services. For example, if a new service is created to enable searches for spinal cord fluid samples across the grid, a service advertisement would be placed in the registry explaining to a service requestor (in an automated way) which data are provided and how to invoke the search. In addition to common metadata that contain generic information about the service provider, metadata based on two standards that reference the data model as registered in the caDSR and the associated semantics as defined in EVS can be supplied:

- **Data service metadata**, which describe the data exposed by the service
- **Analytical service metadata**, which describe supporting operations and associated input and output parameters

caBIG brings together data and tools from more than 50 cancer centers. This number is expected to increase dramatically as the program grows. By requiring each service to describe itself in a manner consistent with caBIG's Common Service Metadata, users who query the central indexing registry service (the Index Service) can discover applications they might be interested in. The Index Service should include details on how precise an application is, how fast it is, and exactly what it provides. The current instantiation of caGrid (Version 0.5) provides a series of high-level APIs and user applications for performing lookups.[37]

Management of these services is facilitated by the Grid Resource and Management (GRAM) service within Globus Toolkit,[38] the open source grid infrastructure toolkit used by caBIG. GRAM provides a Web service interface for initiating, monitoring, and managing the execution of computations on remote computers. The GRAM interface enables a client to specify such characteristics as the type and quantity of resources desired and credentials to be used. GRAM also provides additional operations to monitor the status of computational resources as well as operations to control and monitor the execution of individual tasks. caGrid 1.0, which is expected to roll out in November 2006, will feature a monitoring portal that provides an overview of the status of all services on the grid, along with a geographic map and details of all cancer centers participating in caGrid. If the biomedical community develops a commitment to the caBIG environment, it would be worthwhile to designate responsibility for monitoring GRAM and resolving any issues that arise in terms of performance and adherence to standards.

Users of the grid might want to use multiple services in various sequences to perform calculations (e.g., for protein folding) or to coordinate the processing of specific samples. This choreography of services and the data flows among them can be described in various software formats, but there is consensus within the caBIG community to build workflow definitions based on BPEL, the industry-supported Web service standard with which open source implementations are being developed.[39] BPEL uses an XML format and supports conditional logic, looping, and parallel flows of execution. The choreography of services can be helpful when a series of tasks, such as computations, need to be performed in a specific sequence or if the execution of subsequent tasks will differ, based on the results of the first task. This use of BPEL is similar to providing an executable logic tree that directs the system's components down specific paths in specific situations.

Data exchange mechanisms are another important issue for the performance of the grid. The current architecture specifies using XML as a data exchange mechanism. XML is a commonly accepted Web standard,[40] but it may not be suitable for high-performance data exchange, particularly the exchange of image data, such as microarrays. An oft-cited limitation of XML-based infrastructure is the verbosity of data encoded in XML. This bloat stems primarily from two sources. First, all data are represented textually, even when the data are quantitative or temporal—data types for which more efficient representations are possible. Second, all data are marked with a starting tag and an equivalent ending tag. These tags make the data easily parsable by both humans and machines; however, they create redundancy.

Given this limitation, XML messages need to be compressed whenever resources (e.g., disk space, network bandwidth) are scarce. Several techniques are available, including the ubiquitous Lempel-Ziv algorithm, which is supported by many operating systems and programming languages and is applicable to any textual format, and the XML-specific compression algorithm used by XMill, which has been shown to outperform even ad hoc binary formats. However, XMill is not widely supported (it requires downloading free open source software). Between these two extremes lie additional options.[41] Given this array of options, if caBIG is extended to include XML compression, the architects will need to mandate which approach (or approaches) they expect caBIG participants to support, based on the quality of compression and availability of tools.

New tools are now being developed by vendors, however, and could be evaluated as the architecture evolves.[42] Also, many labs simply download their large data sets to portable media and ship them back and forth via commercial carriers. One researcher commented, "Never underestimate the bandwidth of a FedEx truck." Although this may seem antiquated, it is an expedient approach that may be practical for transferring non-sensitive data.

4. Technical Governance and Organizational Structures

Many of caBIG's information management challenges, such as data interoperability, security and privacy, and use of open source versus commercial off-the-shelf tools, are not limited to the cancer domain. Such issues are found throughout any domain that requires structure and standardization and relies on multiple, disparate development efforts and information sources. With caBIG already in its third year of development, many of these issues have been identified, and working groups have been tasked with developing solutions for them.

Extending the current technology to the entire biomedical community is not a trivial endeavor, however, and the architects will need to answer transition and extensibility questions such as the following:

- How do we facilitate the loose coupling of communities of interest versus creating a more structured global community?
- How will user requirements and development processes be acquired and managed?
- How will open source solutions be maintained?
- How will relationships between internal and external groups involved in the program (governance) be managed?

4.1 Management of Open Source Development

The primary concern with open source systems is a potential lack of stability and the real and perceived danger of decreased technical support compared with the support provided for commercial products. Current caBIG technical support is reported as being responsive and helpful, but it remains to be seen how this type of technical support will be maintained as the program expands beyond NCI's domain. Regarding caBIG use by the biomedical community, it might be difficult to attract significant open source contributions from developers who are not under contract. The most successful open source products, such as Apache and Linux, have been created and developed by software experts who essentially design infrastructures they will personally use (such as an operating system). In caBIG's case, the developers will need to work closely with scientists or clinicians who can explain the requirements and ensure that the open source contribution is valid. Some scientists and clinicians are talented software developers who may be willing to contribute to the program but may not have time to develop the many applications required to fully extend caBIG into the rest of the biomedical space. Therefore, the program may require additional contracted development support and a robust requirements collection and validation process.

4.2 Data Provenance

Establishing data interoperability underpins many of the most critical IT problems in biomedical research. Interoperability requires consistent metadata and data modeling that is based on uniform terminologies and ontologies. The caBIG program is attempting to unify data formats for legacy systems, rapidly evolving scientific knowledge, and software tools still under development.

caBIG components may be used for biomedical research and for patient care. No scientist can afford to have doubts about the validity of the data used in his or her research or for patient care. caBIG's success in the biomedical community will require a well-defined data provenance model so that scientists will have confidence in the results of the research they conduct using data they find via caBIG services or reported on the grid.

A process for data provenance (i.e., tracing and recording the origins of the data and their movement between databases or analysis services) needs to be established to confirm data validity. All data analyses need to be recorded to enable users to use the analyzed data, the raw data, or the data from any point in between. Research results have to be reproducible, and this requirement can be met only if the analysis process is recorded. Finally, the research community is particularly sensitive to intellectual property, and data provenance is essential to understanding data ownership.

The issue of data quality is a related concern. The establishment of metrics that describe the utility of data provided by specific projects will ease some concerns in the community regarding data sharing. Metrics will enable researchers to accept only data that meet a set level of quality. Establishing these metrics for data services on the grid also will encourage researchers to improve and standardize the quality of their data collections.

4.3 The Security and Privacy Model

Security and privacy may be the most critical issues in expanding caBIG use. A high level of security will be required not only because researchers are sensitive about having their data accessed, but also because some of these data are patient specific and may fall under Health Insurance Portability and Accountability Act (HIPAA) guidelines. As caBIG expands across domains, the data sharing and security norms of each community must be accommodated in caBIG's security models. A use of data that may be acceptable to one community may be perceived as inappropriate by another community.

A caBIG contractor completed a security technology evaluation white paper[43] that evaluated a number of candidate technologies in anticipation of a caBIG production release with hundreds of grid services. As outlined by the Architecture workspace in the caBIG community Web site,[44] the requirements described in Table 4-1 will need to be addressed in the expanded caBIG environment.

Table 4-1. Security and Privacy Requirements

Requirement	Description
Secure Communication	The integrity and confidentiality/privacy of messages will be assured.
Authentication	Users will be assured of one another's identity.
Authorization	Resource providers will decide who can access the resources, based on authorization policies.
User/Organizational Attribute Management	There will be a mechanism for caBIG services to request the attributes needed to make an authorization.
Service Delegation	caBIG services will interact with one another on a user's behalf.
Single Sign-On	A single action of authentication will permit access to all services for which a user is eligible.
User/Organizational Management	Organizations will create and manage their own user credentials.
Virtual Organization	Organizations that consist of users from various institutions will be grouped together.

Developing a security model will require input from the technical and research communities. An operations concept model could describe how users would authenticate themselves to the system by obtaining user IDs and passwords or other security tokens. But many technical issues will need to be addressed across the entire biomedical community because many institutions will have to implement the selected security policies.

For example, if the security model requires that a user's institution validate the user's employment and role, the technical means for exchanging and maintaining these data must be worked out. The technical means of allowing independent open source or student developers and users also will have to be considered. Users contributing data for clinical trials and those working at primary care clinics would not be part of any academic institution; therefore, they would not be able to provide an institutional validation. Techniques for ensuring that users are validated and for maintaining secure log-ins will have to be implemented.

Security problems not only will come from external threats, but also from insiders. Internal threats, whether malicious or accidental, are inherently difficult to catch in a research community because much non-threatening exploration takes place.

Security threats, such as Trojan horses, may be maliciously or accidentally included in software contributions. One of the biggest advantages to open source development is that all parties can examine the source code and identify potential attacks. But that assurance requires that someone take the time to perform the inspections. It would be valuable to develop a process for vetting source code before it is incorporated into the software libraries, which is different from compiling and testing. Such examinations also could be helpful in identifying errors (e.g., computation errors) before users have to spend time testing or debugging components.

A multilayer security system that restricts different groups from accessing disparate data sets probably will be needed. Such a system would both satisfy researchers who do not want to share their data with the entire community and address the various interests of the different groups that access the grid (e.g., cancer center staff members, industry, patient advocacy groups). Staged rollout schedules will enable validation of both security and privacy protection technologies and ensure that solutions are scalable.

Addressing where patient de-identification will take place, as required by HIPAA regulations, and developing a governance structure to establish where the fault lies if problems arise, are two additional security issues that will have to be addressed. An audit trail needs to be created to ensure accountability for user actions across the grid.

5. Next Steps

The caBIG model appears, generally, to be extensible to other domains, but it will need to be further developed to include more tools and processes. A certain critical mass of useful data must be exposed on the grid before developers and researchers will invest time in adapting their work to the grid environment. Several scientists commented that a “killer application” that demonstrates the usefulness of the caBIG environment—especially one that leads to an announcement of a breakthrough in a publication such as *Science*—would go a long way toward drawing in new users.

Attention needs to be paid to the human and political aspects of technology adoption and information sharing. Researchers will need to be convinced that sharing data will not compromise the integrity of their studies and that other researchers will not “beat them to the punch” or adversely affect their ability to publish their work. All participants will need to be convinced that data and personal information are secure.

The focus of NCI resources in caBIG development has been, and will continue to be, on cancer. NCI has purposely developed tools that will be useful to other domains, but dedicated resources, both inside and outside the cancer domain and on the caBIG team, will be needed as new communities enter into collaboration.

Another long-term goal is to integrate caBIG with the evolving national electronic health record infrastructure for sharing clinical trial, phenotyping, and other patient data. This access could be the key to creating the “killer application” described previously and could greatly improve the cycle time of the biomedical research process.

5.1 Terminologies and Ontologies

Integrating terminologies and ontologies will continue to be important aspects of integrating new communities into caBIG and will require resources and a governance mechanism. An overall strategy for terminology development, enhancement, and sharing across multiple domains is essential, and terminology curators for different domains will need to be supported.

The development and adoption of open source data standards, such as MIAME, and CDEs for technologies that cross domains should be encouraged. MIAME has sponsorship from industry and participation from both government- and non-government-sponsored researchers. The adoption of a single, industry-wide data standard for a specific technology is more efficient in the long run than mapping among multiple standards.

5.2 Architecture and Development

The caBIG architecture will have to be periodically evaluated as new technologies and standards are developed, particularly those that enhance the SOA and grids. SOA, in particular, is evolving rapidly.

The decision to develop caBIG as an open source effort has lowered the barrier to involvement for many in the community. As caBIG moves toward a true open source community, with developers checking out, modifying, and resubmitting blocks of code, it may make sense to combine this strategy with financial support for some management areas, such as documentation and integration. It may also be worthwhile to consider identifying a champion for the caBIG effort in the long term who will play a role similar to the one Linus Torvalds plays in the Linux community. Torvalds not only initiated Linux development, he also maintains leadership of the open source effort, attracting skilled developers and ensuring that the work products meet the overall architectural and technical goals.

The following steps could be taken to leverage open source resources:

- Define needs in the most generic manner possible, avoiding, for example, the use of highly customized terminology that obscures underlying similarities.
- Perform a thorough search of global open source resources for existing tools or components that meet many or all generic needs.
- Evaluate the maturity of candidate open source components by using, for example, Bernard Golden's Open Source Maturity Model.[45] Earlier evaluations may exist and new ones should be captured for reuse by other caBIG developers.
- Direct developers to participate in caBIG development groups rather than working in isolation. This type of goal-oriented participation generally requires two components:
    1. A willingness to help the caBIG community raise the tool's overall quality and capabilities (which is highly compatible with the caBIG community's goals)
    2. Contributions of new features in ways that are compatible enough with the existing code base (e.g., extensions rather than changes to existing interfaces) and modular enough (e.g., easily removed if not wanted) to be acceptable to the community
- Harvest the resulting generalized products and apply them to the needs of the sponsoring community. The process of applying the products usually entails the addition of a relatively small amount of unique, domain-specific code that achieves the customization that otherwise would have been scattered throughout the generic product if an open source approach had not been used. Ideally, the custom code will take the form of purely declarative data, such as data in a configuration file.
- For long-term support, continue contracting for participation in the open source software support process. Compared with full internal support for custom software, the level of support needed for open source software typically is much lower and can be shared across a suite of open source products. Another advantage is that developers in such support roles tend to be both happier and more effective because they are not constantly dealing with the minutia of bugs and obsolete code and are tied into a community of experts in the code they are supporting.

It would be valuable to consider funding a systems integrator at a centralized site who will take responsibility for organizing and supporting the open source community and accomplish the following:

- Provide a structured, protected test environment for testing components against a well-understood baseline
- Manage the performance, monitoring, and security of the grid
- Package the components, along with appropriate documentation and tools, in a form that is easy to implement

- Provide help desk and developer support, which will be particularly important because tools (e.g., caTISSUE) are planned to be used to support clinical patient care in healthcare delivery organizations

One way that caBIG could reduce integration risks is to develop a researcher-oriented universal data access strategy with a specific tool that starts by providing only large-granularity, minimal-automation access to remote sites, but that can, by design, accommodate incremental increases in both the level of detail (decreasing granularity) and the degree of data automation at each site. The tool should not be costly or complex and should be easy to use. A mental model of the ease of use (but not of the incremental growth ability) would be something as minimal as a "caGoogle" tool that performs simple but smart keyword searches and may provide results that are as simple as clinic names and phone numbers of researchers.

5.3 Certification

Certification of services is an essential, maturing process. caBIG developers may want to leverage the work of other domains, such as the Defense Information Systems Agency Federated Development and Certification Environment (FDCE). FDCE focuses on addressing challenges that result from the development and certification of net-centric services and on providing policies, processes, and infrastructure that enable services to be refined, tested, evaluated, and certified under increasingly rigorous circumstances until operational deployment is achieved.

6. Conclusion

The alternative to expanding caBIG across all domains is to develop an equivalent tool set multiple times. The cost of doing nothing will result in the expenditure of millions of dollars in the development of still more silos that impede the efforts of the cancer and non-cancer communities. There are no inexpensive or easy answers, but there is enough commonality across the domains to make the expansion of caBIG worth pursuing.

Acronyms

AERS	Adverse Event Reporting System
API	Application Programming Interface
BPEL	Business Process Execution Language
caBIG™	Cancer Biomedical Informatics Grid™
caBIO	Cancer Bioinformatics Infrastructure Object
caDSR	Cancer Data Standards Repository
caTISSUE	Cancer Tissue Database
CDE	Common Data Element
CTSA	Clinical and Translational Science Awards
EEG	Electroencephalogram
EVS	Enterprise Vocabulary Services
FDCE	Federated Development and Certification Environment
FFRDC	Federally Funded Research and Development Center
GRAM	Grid Resource and Management
HIPAA	Health Insurance Portability and Accountability Act
ICD	International Classification of Diseases
IT	Information Technology
MDA	Model Driven Architecture
MedDRA	Medical Dictionary for Regulatory Activities
MIAME	Minimum Information About a Microarray Experiment
MRI	Magnetic Resonance Imaging
NCI	National Cancer Institute
NCRR	National Center for Research Resources
NHLBI	National Heart, Lung, and Blood Institute
NIH	National Institutes of Health
NINDS	National Institute of Neurological Disorders and Stroke
OWL	Web Ontology Language
RDF	Resource Description Framework
REMBRANDT	Repository for Molecular Brain Neoplasia Data
SDO	Standards Development Organization

SELDI	Surface-enhanced Laser Desorption Ionization
SOA	Service Oriented Architecture
UML	Unified Modeling Language
UMLS	Unified Medical Language System
VCDE	Vocabularies and Common Data Elements
VISDA	Visual and Statistical Data Analyzer
XML	Extensible Markup Language

Contributors

This report was prepared by a team of MITRE staff members. Team members are as follows:

- Terry Bollinger, M.S.
- Steven Decker, M.S.
- Don Faatz
- Brandon Higgs, Ph.D.
- Kathy Lesh, R.N., Ed.M., M.S., B.C.
- Joseph Mitola, Ph.D.
- Olivia Peters, M.S., Project Lead and lead author
- Robert Mikula, M.S.
- Peter Mork, Ph.D.
- Matthew Seguin, M.S.
- Jean Stanford, Project Manager
- Gary Vecellio, M.S.
- Marion Warwick, M.D.

Endnotes

[1] The MITRE Corporation, *caBIG™ Overview*, May 2006, http://www.ncrr.nih.gov/CRinformatics/caBIG.pdf. For more information on caBIG, go to https://cabig.nci.nih.gov.

[2] "The IBM Linux Technology Center is a worldwide organization with teams in approximately 40 locations. It comprises some 600 engineers worldwide, of whom more than 300 work full-time on Linux as part of the open source community. The investment to expand Linux development in Brazil complements ongoing work at IBM's Linux Integration Centers, Linux Innovation Centers, and Linux Competency Centers, all of which help customers port applications to Linux." In "IBM Expands Linux Technology Center in Brazil," *The IBM LinuxLine*, May 24, 2006, http://www.dbta.com/linuxline/archives/5-24-06.html.

[3] "Metadata" is a term commonly used in the biomedical informatics field: "(1) Information about a data set which is provided by the data supplier or the generating algorithm and which provides a description of the content, format, and utility of the data set. Metadata provide criteria which may be used to select data for a particular scientific investigation. (2) Information describing a data set, including data user guide, descriptions of the data set in directories, and inventories, and any additional information required to define the relationships among these." See glossary at http://podaac.jpl.nasa.gov/glossary.

[4] Wolfgang G, *DOT-COMing the GRID: Using Grids for Business*, Sun Microsystems, Inc., http://www.sun.com/software/gridware/article.xml.

[5] Foster I, Kesselman C (editors), *The Grid: Blueprint for a New Computing Infrastructure*, first edition, Morgan Kaufmann Publishers, November 1998.

[6] Kher M, Oster S, "caGrid Version 0.5 Architecture Overview Advertisement and Discovery," caBIG Architecture Workspace Face to Face, Georgetown University, August 2005.

[7] Contie VL, "Clinical and Translational Science: Speeding the Translation of Medical Discovery into Enhanced Patient Care," *NCRR Reporter*, Winter 2006, http://www.ncrr.nih.gov/newspub/winter06rpt/stories1.asp.

[8] See, for example, the activities discussed at http://www.nhlbi.nih.gov/meetings/workshops/population.htm.

[9] See http://rembrandt.nci.nih.gov.

[10] See http://gforge.nci.nih.gov/projects/lexbig.

[11] This section is based on site visits by MITRE staff to the Mayo Clinic and interviews with key leaders concerning their use of caBIG across the clinic. The site visits occurred in May 2006.

[12] This section is based on interviews by MITRE staff with caBIG participants from the University of Pennsylvania during the caBIG annual conference in February 2006 and immediately thereafter.

[13] See http://discover.nci.nih.gov/gominer.

[14] The Gene Ontology project provides a controlled vocabulary to describe gene and gene product attributes in any organism. See http://www.geneontology.org.

[15] This section is based on interviews by MITRE staff with Georgetown University staff in February and March 2006.

[16] NIH Roadmap Investigator Meeting: Inventory and Evaluation of Clinical Research Networks and Re-engineering the Clinical Research Enterprise, National Institutes of Health, Bethesda, Maryland, May 30, 2006, https://vishnu.cceb.upenn.edu/portal/page?_pageid=36,63082,36_69934&_dad=portal&_schema=PORTAL.

[17] NCRR Clinical Trials Networks Contract, Duke Clinical Research Network, Robert A. Harrington, M.D., P.I.

[18] University of Pennsylvania School of Medicine, J. Richard Landis, Ph.D., P.I.

[19] The Electronic Primary Care Research Network, ePCRN, University of Minnesota, Kevin Peterson, M.D., P.I.

[20] Reported at the NIH Roadmap Investigator Meeting: Inventory and Evaluation of Clinical Research Networks and Re-engineering the Clinical Research Enterprise, National Institutes of Health, Bethesda, Maryland, May 30, 2006, https://vishnu.cceb.upenn.edu/portal/page?_pageid=36,63082,36_69934&_dad=portal&_schema=PORTAL.

[21] For documentation on caMassClass, see http://rss.acs.unt.edu/Rdoc/library/caMassClass/doc.

[22] R is an open source statistical analysis package that is widely used in the biomedical research community.

[23] For more information on MDA, see http://www.omg.org/mda.

[24] Many large-scale projects have been undertaken in the classified community that may be relevant, but they may not be cited here.

[25] For more information on MDA, see http://www.omg.org/mda.

[26] The MITRE Corporation, *caBIG™ Overview*, May 2006, http://www.ncrr.nih.gov/CRinformatics/caBIG.pdf.

[27] See discussion of MDA at https://cabig.nci.nih.gov/workspaces/VCDE/Documents/Useful_Presentations/Model%20Driven%20Architecture.

[28] See http://www.uml.org for an overview of UML.

[29] See http://www.nlm.nih.gov/research/umls.

[30] See http://www.nlm.nih.gov/research/umls/license.html.

[31] Golbeck J, Fragoso G, Hartel FW, Hendler J, Oberthaler J, Parsia B, “The National Cancer Institute’s Thesaurus and Ontology,” *Journal of Web Semantics* 1:75–80, 2003.

[32] “OWL is a Web Ontology language. Where earlier languages have been used to develop tools and ontologies for specific user communities (particularly in the sciences and in company-specific e-commerce applications), they were not defined to be compatible with the architecture of the World Wide Web in general, and the Semantic Web in particular. OWL rectifies this by providing a language which uses the linking provided by RDF to add the following capabilities to ontologies:

- Ability to be distributed across many systems
- Scalable to Web needs
- Compatible with Web standards for accessibility and internationalization.
- Open and extensible”

See http://www.w3.org/2003/08/owlfaq.html.

[33] For more information on Protégé, see http://protege.stanford.edu.

[34] See http://gforge.nci.nih.gov/projects/lexbig.

[35] An ontology is “a domain of discourse for describing some reality. A set of concepts, the attributes of these concepts and the relationships among concepts that characterize a given application area.” From van Bemmel JH, Musen MA (editors), *Handbook of Medical Informatics*, Springer-Verlag, 1997, p. 587.

[36] Flournoy R, Lee E, Mikula R, “Testing Net-Centric Systems of Systems: Applying Lessons Learned from Distributed Simulation Testing,” NDIA Systems Engineering Conference, October 2005.

[37] The MITRE Corporation, *caBIG™ Overview*, May 2006, http://www.ncrr.nih.gov/CRinformatics/caBIG.pdf.

[38] Foster I, “Globus Toolkit Version 4: Software for Service-Oriented Systems,” *IFIP International Conference on Network and Parallel Computing*, Springer-Verlag LNCS 3779, 2005, pp. 2–13, http://www.globus.org/alliance/publications.

[39] Fox G, Ho A, Pierce M, “DoD Grid Opportunities and Technology Overview,” Community Grids Laboratory, Indiana University, and Anabas Inc., August 2005.

[40] See http://www.w3.org.

[41] Cokus M, Winkowski D, "XML Sizing and Compression Study for Military Wireless Data," Proceedings of the XML Conference and Exposition, Baltimore, Maryland, December 8–13, 2002, http://www.idealliance.org/papers/xml02/dx_xml02/papers/06-02-04/06-02-04.html.

[42] Coffee P, "SOA Still Gaining Momentum," *eWeek*, June 12, 2006.

[43] caBIG™ Security Technology Evaluation White Paper, October 7, 2005, https://cabig.nci.nih.gov/workspaces/Architecture/Security_Tech_Eval_White_Paper_Provisional.

[44] See https://cabig.nci.nih.gov/workspaces/Architecture/Documents/Arch_Workspace.

[45] *The Open Source Maturity Model*, Navica Inc., http://www.navicasoft.com/pages/osmmoverview.htm.