

DRAFT

caBIG Workspace Developer Project Form

Developers, please complete this form in advance of the caBIG kickoff meeting and return by e-mail to adamsm@mail.nih.gov. Completed forms will be made available to all participants in advance of the meeting to enhance workspace discussions. During our conversations with you, we expressed the aspect of your program that we would like you to develop in the first year of the caBIG pilot; it is this we are asking you to address - here and in your presentation.

1. Sponsoring Cancer Center

Holden Comprehensive Cancer Center, The University of Iowa.

2. Workspace

Integrative Cancer Research

3. Project or Activity

- I. Transcript Annotation Prioritization Screening System (TrAPSS)
- II. Clinical and Expression Database (CED)
- III. Integrated Expression Environment (IEE)
- IV. Custom Sequence Annotation (CSA)
- V. Genotyping Management System (GenoMap)

4. Workspace needs the project meets

Needs	Projects				
	TrAPSS	CED	IEE	CSA	GenoMap
Clinical data management tools and databases.		X			*
Distributed general data sharing and analysis tools	X	X	X	X	X
Translational research tools.	X	X		*	
Access to data.	X	X	X		X
Common Data Elements (CDE) and architecture.	*	*	*	*	
Vocabulary and ontology tools and databases.			*		
Visualization and front-end tools	X	*	X	*	X
Microarray and gene expression tools		X	X		
LIMS	X				X
Database and datasets		X	X		

X – yes, * - to a lesser degree

5. Stage of project maturity (Conceptual, early beta, regular end-user use at parent center, regular use in the community)

TrAPSS: second release currently being deployed with extensive user testing and feedback on first release.

CED: early beta; additional functionality currently being implemented

IEE: regular end-user use at parent center

CSA: extensive end-user use at parent center (approximately 2,000,000 sequences submitted to GenBank); deployed at other institutions.

GenoMap: regular end-user use at parent center

6. Technical details of Tools

a. Software Architecture (These will likely be preliminary)

i. System design

ii. Component details

iii. Relevant standards

iv. UML schematics (if valid)

v. Size of project installed software base

b. Development Environment (tools, languages, bug tracking, etc.)

TrAPSS:

a.i. System Design

TrAPSS is centered around expressed transcripts within a genome context. "Target" genes for mutation screening are entered into the system. The gene structure, flanking genomic sequence, and associated sequence features and annotation are automatically acquired from Ensembl and cached in a local database for analyses. A custom algorithm that quantitatively assesses gene-based annotation to infer pathogenicity and guides screening assays.

a.ii. Component Details

Univ. of Iowa components:

- PHP-web interface
- PHP modules (database interface)
- Java modules (database interface)
- Perl modules (database interface)
- Driver/GREEN (Java client interface)
- PrimerViewer (Java visualization, analysis, and primer generation/selection)
- Primer3 server (Java)
- Populate (Perl data acquisition with Ensembl)
- PrimerManager (Java assay/primer management tool)
- SSCPTool (Java and web-based results acquisition and storage)
- Juxtapositron (Java interface for high-throughput sequence-based mutation evaluation and visualization)

Community components:

- Apache
- MySQL
- Primer3
- Ensembl modules
- BLAST
- BLAT

a.iii. Relevant Standards:

Ensembl modules

Locally developed modules modeled after Ensembl modules.

a.iv. UML schematics (if valid)

None.

a.v. Size of project-installed software base

Approximately 10 users plus a number of users of subcomponents.

b. Development Environment (tools, languages, bug tracking, etc.)

Apache, MySQL, PHP, Perl, Java, and Ensembl Perl modules. PHP for web interface, and Java/WebStart for application interfaces.

CED:

a.i. System Design

The Clinical and Expression Database (CED) integrates both clinical data (patient data) with expression data (currently microarray and EST data), for analyses and annotation utilizing: UniGene, LocusLink, GO, EC numbers, pathways, and genomic mapping information. This facilitates clinical and expression "synergy" by enabling queries for genes and tissue samples based on clinical and expression data.

a.ii. Component details

Webserver

Database

PHP interface and data loading tools.

a.iii. Relevant standards

MIAMI, MAGE-ML

a.iv. UML schematics (if valid)

None.

a.v. Size of project-installed software base

One base install – testing only.

b. Development Environment (tools, languages, bug tracking, etc.)

Apache, postgres, PHP.

IEE:

a.i. System Design

The Integrated Expression Environment (IEE) is web-based data management system for expression-based technologies (Affymetrix gene chips, printed glass slides, SAGE, MPSS, etc.) that enables the sharing and distribution of data, annotation, and results between geographically separated investigators. Microarray experiments (with annotation) are loaded into the system. Specific analyses may be automated for availability to the broad range of data within the system.

a.ii. Component details

Webserver

Database

PHP interface

Perl for annotation acquisition and generation.

Affymetrix web-site (for annotation)

a.iii. Relevant standards

MIAMI, MAGE-ML

a.iv. UML schematics (if valid)

None.

a.v. Size of project-installed software base

One install base, multiple on-site users.

b. Development Environment (tools, languages, bug tracking, etc.)

Apache, Perl, Java, XML/MAGEML, MySQL,

CSA:

a.i. System Design.

Sequences are loaded, and a series of annotation analyses are specified. The resulting computation(s) are executed and stored in a database. Results are presented via a web server. Currently implemented components include sequence extraction (phred; Ewing et al, 199x), feature annotation (ESTprep; Scheetz et al. 2003), sequence homology (BLAST; Altshul et al. 199x), sequence clustering (UIcluster; Trivedi et al. 200x), sequence assembly (phrap; ??).

a.ii. Component Details

phred
ESTprep
BLAST (with Bioperl)
UIcluster
Phrap
Database
webservice

a.iii. Relevant standards

FASTA, BioPerl

a.iv. UML schematics

(none)

a.v. Size of project-installed software base

1 software base (multiple users) + components distributed to multiple off-campus sites.

b. Development Environment (tools, languages, bug tracking, etc.)

Perl, C, MySQL, Apache, Linux

GenoMap:

a.i. System Design

Patient/clinical data are entered for the purpose of genotyping. Marker data from Marshfield is loaded. Digitized images of genotyping gels are read with custom, automated genotype reading software. Genotypes are validated by redundancy, for Mendelian consistency, and stored. Formatted linkage files may then be exported for analyses.

a.ii. Component details

GenoMap (web interface to tools)
GenoScape (C-based automated genotype-calling application)
SubjectLog (Java patient entry)
MarkerLog (Java marker management)
Verification (Java genotype comparison and Mendel check)
ExperimentEditor (Java experiment design tool)
Lanalysis (Java linkage analysis data exporter and formatter)
Servers for applet client/server communication
Database
webservice

a.iii. Relevant standards

Linkage format

a.iv. UML schematics (if valid)

None.

a.v. Size of project-installed software base

One installation, 6 users.

b. Development Environment (tools, languages, bug tracking, etc.)

C, Java, Sybase, SQL

7. Does the project make use of existing standards? If so, what are they? (e.g. bioinformatics standards such as MIAME for microarrays, or software standards such as XML)

Yes, see #6 above for each project.

8. Does other software in the community meet this need? Is this software open source? Can it be harnessed?

CED: Unique software that can be harnessed. Open source.

TrAPSS: Unique software that can be harnessed.

IEE: Some unique software that can be harnessed with other software in community available. Open source.

CSA: Unique software that can be harnessed with integration of pre-existing components. Open source.

GenoMap: Some unique software that can be harnessed with other commercial software in community. Open source.

9. Points of possible interoperability with other caBIG systems (This might include communication with other caBIG databases, use of caCORE APIs, caBIG-compatible APIs, etc.)

We are enthusiastic about participating with caBIG interoperability as soon as decisions about projects are made and specifications for such modules become available (such as adopting the caBIO). However, since we cannot predict what standards will be adopted by caBIG participants, for this section of the document we assume that the caBIO will be adopted as a basis for interoperability.

TrAPSS: integration with site-specific data/databases for locally generated expression data for candidate prioritization. Interoperability for expression data between TrAPSS and other caBIG modules would be achieved with the caBIO/(SOAP) API using PERL. Note, TrAPSS already has similar interoperability with Ensembl so utilizing the caBIO would be a modification of existing modules.

CED: high potential for interoperability with: clinical data, expression data, and annotation. The Clinical and Expression Database would achieve interoperability with either the caBIO Java API and SOAP API with Perl.

IEE: high potential for interoperability for expression data, annotation, and analyses. The Integrated Expression Environment would achieve interoperability of expression and annotation

data with the caBIO (Java and SOAP APIs). However, for analyses additional APIs would need to be developed for caBIG interoperability.

CSA: already has high degree of interoperability between components. Interoperability with caBIG would require the development of import and export modules consistent with the caBIO (utilizing either the Java or SOAP APIs).

GenoMap: stand-alone system for genotyping.

10. What resources are proposed to achieve caBIG interoperability?

- a. Developmental requirements**
 - i. Software (re)engineering**
 - ii. Standards adoption**
 - iii. Platform migration**
- b. Infrastructure**
 - i. Facilities**
 - ii. Management tools**
 - iii. Personnel**

TrAPSS:

a.i. Software (re)engineering

Currently finalizing second complete implementation. The data interfaces may need to be modified to be compatible with other caBIG modules. The caBIO Data Access Objects may be rapidly adopted by developing new TrAPSS modules compatible with the Data Access Objects.

a.ii. Standards adoption

Standards for data exchange will need to be adopted (based upon caBIG community needs) and implemented.

a.iii. Platform migration

Platform independent (Java Webstart; Windows, Linux, and Mac). Server-side applications currently run under Linux.

b.i. Facilities

b.ii. Management tools

b.iii. Personnel

3 staff programmers, and 4 students.

CED:

a.i. Software (re)engineering

Additional software will need to be implemented to be compatible with other caBIG modules.

a.ii. Standards adoption

Standards for data exchange will need to be adopted and implemented.

a.iii. Platform migration

Web-based interface is inherently portable. Current servers are on Linux platform.

b.i. Facilities

b.ii. Management tools

b.iii. Personnel

2 staff programmers, and 3 students.

IEE:

a.i. Software (re)engineering

Modifications will need to be made to be compatible with other caBIG modules. Implemented analyses will need to be expanded.

a.ii. Standards adoption

Currently, IEE supports data exchange in several formats, including the chip composition and expression data formats from Affymetrix (CEL, CHP, DAT), the GPR format (expression data), ?? (spot position), and MAGEML formats. Additional standards for data exchange and interoperability will need to be adopted and implemented as adopted by the caBIG community.

a.iii. Platform migration

Web-based interface is inherently portable. Current server-base is Linux. The majority of the developed code is implemented in Perl and PHP.

b.i. Facilities

b.ii. Management tools

b.iii. Personnel

2 staff programmers, and 3 students.

CSA:

a.i. Software (re)engineering

Additional software will need to be implemented to be compatible with other caBIG modules. Existing packages (e.g., ESTprep) could be modified to encompass more generic forms of annotation.

a.ii. Standards adoption

Standards for data exchange will need to be adopted and implemented (upon adoption by the caBIG community). All sequence and quality data is currently maintained in the FASTA format.

a.iii. Platform migration

Many of the components of CSA are applications previously developed by the community (e.g., phred, phrap). Licensing for platform migration may not be feasible. Currently implemented in C and deployed on Linux.

b.i. Facilities

b.ii. Management tools

b.iii. Personnel

2 staff programmers, and 2 students.

GenoMap:

a.i. Software (re)engineering

A large portion of the system software will need to be ported to a more recent version of Java. GenoMap will be reimplemented in Java to enable cross-platform portability and compatibility.

a.ii. Standards adoption

All identifying information is removed prior to entry into the system. Exporting of genotypes is currently supported using themlink format.

a.iii. Platform migration

Most of the GenoMap components are written in Java and are thus platform independent, given the Java runtime environment (Windows, Macintosh, and Linux/UNIX). The GenoScope component is currently implemented in C/Xwindows and thus requires an Xserver to run.

b.i. Facilities

b.ii. Management tools

b.iii. Personnel

3 staff programmers, and 2 students.

11. Draft 12-month work plan, with milestones to achieve caBIG interoperability.

TrAPSS:

Month 1) Design modules to communicate with other caBIG modules

Month 2) Design target scoring system (TSS).

Month 3) Implement communication modules

Months 4/5) Implement TSS

Month 6) Testing

Months 7/8) Deployment, user testing and feedback

Months 9/10) Bug fixes and feedback driven modifications

Months 11/12) Testing, validation, and deployment of final product.

CED:

Month 1) Design modules to communicate with other caBIG modules

Month 2) Implement communication modules

Month 3) Implement additional required functionality for new data sources/types.

Month 4) Test and validation.

Months 5/6/7) Deployment, user testing and feedback

Months 7/8) Refinement from user feedback

Months 9/10) Bug fixes and redeployment of final version.

Months 11/12) Testing and validation.

IEE:

Month 1) Design modules to communicate with other caBIG modules

Month 2) Design modules to accommodate other sources of data.

Months 3/4) Implement communication modules and other data source modules

Months 5/6/7/8) Deployment and training. Load data from multiple sites for testing and sharing.

Month 9) Design and implement automated analyses modules based on user feedback and needs.

Months 9/10) Bug fixes and feedback driven modifications.

Months 11/12) Testing, validation, and deployment of final product.

CSA:

Month 1) Design components to communicate with other caBIG modules
Month 2) Redesign components to be more generalizable and simpler to use.
Month 3) Redesign and implement ESTprep to address general sequence features.
Months 4/5/6) Testing and validation.
Months 7/8/9) Deployment, training, user testing and feedback
Months 10/11/12) Testing, validation, and final software fixes.

GenoMap:

Months 1/2) Redesign components to interface with caBIG components/projects.
Months 2/3/4/5) Port existing tools to recent version of java.
Months 5/6/7) Testing
Month 7/8) Design and implement modules to communicate with other caBIG modules.
Months 7/8/9) Deployment, training, user testing and feedback
Months 9/10) Bug fixes and feedback driven modifications
Months 11/12) Testing, validation, and deployment of final product.