Putting it All Together: Complete Architectures
PUTTING IT ALL TOGETHER: COMPLETE ARCHITECTURES

- The work flow of creation
- An example - - caBIO
The workflow of application creation

• Problem statement
• Component definition
• Architecture
  ➢ Ease of creation
  ➢ Robustness and maintainability
  ➢ Performance and security
• Connecting the components
• Testing: use cases
The National Cancer Institute
Center for Bioinformatics
ncicb.nci.nih.gov

The Mission Statement (abbreviated). The NCICB coordinates and deploys informatics in support of NCI research initiatives. Its goal is to maximize interoperability and integration of NCI research and its related information. Informatics support provided by the NCICB includes platforms, services, tools, and data. The NCICB establishes information technology standards (both within and outside of NCI).
caCORE and the NCI information components

- caCORE is the information engine

Courtesy of National Cancer Institute.
Cancer Bioinformatics Infrastructure Objects (caBIO)

caBIO details: see for example http://open-bio.org/bosc2002/slides/2002-08-01-Covitz-caBio.ppt

Courtesy of National Cancer Institute.
The Genomic Past

- Little or no data integration
- Lack of standards
- Little or no software code re-use
- Duplication of efforts
- Duplication of data
- Lack of common vocabulary
- Maintenance nightmares
caBIO Overview

- caBIO is a standards based set of genomic components
- caBIO objects simulate the behavior of actual genomic components such as genes, chromosomes, sequences, libraries, clones, ontologies, etc.
- caBIO objects provide access to a variety of genomic data sources including GenBank, Unigene, LocusLink, Homologene, Ensemble, GoldenPath, and NCICB’s CGAP (Cancer Genome Anatomy Project) data repositories
- caBIO is “open source”
Requirements Process

- **High Level Use Cases**
  - Initial high level Use Cases were developed for gene components
  - Additional high level Use Cases were added as additional functional areas are mapped (e.g. pathways, therapies, microarray objects, mouse models)
  - Detailed Use Cases are derived from working with domain experts in requirements gathering.

Figure by MIT OCW.
Architecture

- The core infrastructure exhibits an n-tiered architecture with client interfaces, server components, back-end objects and data sources
- Clients (browsers, applications) can receive information (HTML and XML) from back-end objects over HTTP
  - Client applications can also communicate with back-end objects via Java RMI (Java applications)
  - Non-Java based applications will communicate via SOAP
- Server components communicate with back-end objects via Java RMI
- Back-end objects communicate directly with data sources (database, URLs, flat files)
- A UDDI registry will be configured to advertise services
  - RDF is currently used to advertise services to crawlers and agents
The complete architecture ... caBIO

Courtesy of National Cancer Institute.

NCICB National Cancer Institute © cfdewey 2003
Architectural components

- **Client Technology**
  - **Industry Standard Web Browsers** - Netscape 4+ and IE 4+
  - **Java Applications** – Applications that implement the Java programming language, an object-oriented language which provides portability and many other features
  - **Non-Java Applications** - Applications usually written in Perl, C, etc.
    - Non-Java applications use SOAP clients to interface with caBIO
      - SOAP:Lite for Perl – A collection of Perl modules which provides a simple and lightweight interface to the Simple Object Access Protocol (SOAP)
  - **Agents** - Software programs that perform a function for a user in a trusted fashion, can learn or be taught its function, and can perform actions for the user without permission
  - **RDF** – The Resource Description Framework is a foundation that advertises Web services via the Web. RDF is used to describe the content and services available at a particular Web site. (passive)
  - **UDDI** – Universal Description, Discovery and Integration is a foundation to enable businesses to discover and transact business with each other using preferred applications (active)
Presentation Layer Technology

- **Jakarta Tomcat** - Servlet+JSP Engine which is a subproject of the Jakarta Project
- **JSPs** - Java Server Pages are web pages with Java embedded in the HTML to incorporate dynamic content in the page
- **Java Servlets** – Server-side Java programs that web servers can run to generate content in response to client requests
- **Java Beans** – Reusable software components that work with Java
- **XML** – The Extensible Markup Language is a universal format for structured data on the Web
  - XSL/XSLT – The Extensible Stylesheet Language is a language for expressing stylesheets. XSL Transformations (XSLT) is a language for transforming XML documents
  - XLink – The XML Linking Language allows elements to be inserted into XML documents in order to create and describe links between resources
  - DOM – The Document Object Model is a platform and language independent interface that allows programs to dynamically access content, structure, and document style
Presentation Layer Technology (cont.)

- **SOAP** – The Simple Object Access Protocol is a lightweight XML based protocol for the exchange of information in a decentralized, distributed environment. It consists of an envelope that describes the message and a framework for message transport.
  
  - **Apache SOAP** – An implementation of the W3C SOAP specification. The Apache SOAP provides a server-side infrastructure for deploying, managing, and running SOAP enabled services.
Architectural components (contd.)

Presentation Layer

- The Mediator Servlet manages the user session, forwards requests to the JSP, and returns HTML
- JSPs forward requests to the UI Bean, and returns XML or HTML to the client
- The UI Bean receives the a Domain Object and converts the object to XML or HTML
- The DOM Writer performs the Conversion to XML while the Conversion Servlet represents the Domain Object as a Document Object (DOM)
  - The conversion is performed by calling the Domain Objects toXML() method
Architectural components (contd.)

- **Object Layer**
  - Object Managers are created to implement complex scientific concepts
  - The Role Model object verifies user permissions to the data if applicable
  - The Remote Manager Interface abstracts the RMI layer
    - Allows RMI to be easily replaced by EJB or other communication venues
  - Domain Objects are serialized and XML enabled

Courtesy of National Cancer Institute.
Architectural components (contd.)

- **Object Layer Technology**
  - **Java Applications/Objects** – Applications/Objects that implement the Java programming language, an object-oriented language which provides portability and many other features.
  - **Java RMI** – Remote Method Invocation is distributed computing in Java. It is a simple technology for object communication that allows remote objects to act as local Java objects.
  - **JDBC** – Java Database Connectivity is a Java API for executing SQL statements and connecting to databases.
  - **SOAP** – Simple Object Access Protocol extends object interoperability to other programming languages (Perl, Python, C++)
  - **DAS** – The Distributed Annotation System is an emergent system for retrieving genomic annotations from a variety of data sources (GoldenPath, Ensembl)
Architectural components (contd.)

- **Data Access Objects**
  - Provides the object relational mapping
  - Database independent persistence
  - Allows for the introduction of new data sources
  - Abstracts persistence details from the domain objects
  - Allows for federated database topology

Courtesy of National Cancer Institute.
The complete architecture ... caBIO

Clients
- Browsers
- External Java Apps
- Agents
- Other Apps
- Internal Java Apps

Presentation Layer
- Web Server
  - Servlet Container
  - JSPs
  - Servlets
  - SOAP Engine
- UI Bean
  - XML Builder
  - XSLT Engine
- DTDs
- XSL Style Sheet

Object Layer
- Object Managers
- Data Access Objects
- RMI

Data Sources
- Databases
  - JDBC
- HTTP
  - URLs
- FTP
  - Flat Files

Domain Objects
- Genes
- Chromosomes
- Tissues
- Clusters
- Sequences
- Libraries
- Diseases
- Other

Courtesy of National Cancer Institute.

NCICB National Cancer Institute © cfdewey 2003