PROTEOMICS LABORATORY INFORMATION MANAGEMENT SYSTEM

BY
OLGA TCHUVATKINA - BIOINFORMATICS
THE FOX CHASE CANCER CENTER

ARCHITECTURE DOCUMENT

DRAFT

DOCUMENT NUMBER: v1.4
CREATED: 10/19/2004
LAST UPDATED: 03/28/2005
**TABLE OF CONTENTS**

OVERVIEW .................................................................................................................. 2  
COMPONENT DIAGRAMS .......................................................................................... 3  
DOMAIN OBJECT MODEL ......................................................................................... 5  
SOFTWARE DEVELOPMENT PROCESS .................................................................... 6
**OVERVIEW**

*Figure 1.0 - ProtLIMS architecture overview*

**Presentation Tier**
- **Web interface**
- **Struts**
- **Business Facade**

**Business Tier**
- **Infrastructure Session Beans (Basic logic)**
- **Core Session Beans (Business logic)**
- **Entity Beans**
- **Data Access Objects**

**Data Tier**
- **Files**
- **Database**

**Entity beans** are used for maintenance of sample-data, laboratory-project registry and inventory tables in database.

**Data Access Objects** (DAO) are used to retrieve data for browse. DAOs work with database views to obtain big subsets of data for read only.

**Infrastructure Session Beans** perform maintenance on entity beans.

**Core Session Beans** use Infrastructure Session Beans, Entity Beans and DAO to provide business logic methods.

**Business Façade** is used by Presentation tier to access Business tier.

**Web interface** uses struts framework to process user requests.
**COMPONENT DIAGRAMS**

Three main scenarios for system usage: maintain data, browse data, maintain lookup data.

Maintenance scenario is used to manage complicated data structures. Almost all business logic methods require updates in multiple objects from different packages. Core Session Beans level provides business methods to handle these updates.

*Figure 2.0 – Maintenance scenario*

Example: user registers new datafile as procedure output. This action requires updates at least in datafile (sampledata package), limsfile (inventory package) and procedure (labproject package). Infrastructure session beans provide methods to work with each object separately; core session bean uses infrastructure session beans and registry entity beans to perform complicated update action. Web code uses business façade to access business logic methods on core session bean.
Admin scenario is used to manage look-up like data. Update for this type of data doesn’t require use of session beans.

*Figure 3.0 – Admin scenario*

Browse scenario uses DAO to access database for read only purposes. Browse doesn’t require use of beans – in this case DAO is more efficient way to access database.

*Figure 4.0 – Browse scenario*
**DOMAIN OBJECT MODEL**

**Domain object model** is separated into four packages: sampledata, labproject, inventory and registry

Sampledata package contains objects for different types of samples and data such as generic sample and datafile, raw sample, 2d gel sample, gel images etc.

Labproject package contains project scope objects: laboratory, project, procedure, process, protocol etc.

Inventory package contains assorted objects related to inventory management: equipment, storage devices, files in storage, sample plates etc.

Registry package contains association objects such as:
- ProjectSample connects Project and Sample many to many
- ProjectDataFile connects Project and DataFile many to many
- DataFileLimsFile connects Data and File many to many
- ProjectPersonnel connects Project and LIMS many to many etc.

See domain class diagrams for details
SOFTWARE DEVELOPMENT PROCESS

ProtLIMS Iterative Software Development Process Approach

ProtLIMS development process starts from high-level system analysis. Development process iteration goes from business and architectural requirements analysis to system implementation through application modeling. ProtLIMS testing in the end of iteration cycle results in new requirements and new iteration cycle.

Figure 5.0 – System Architecture Methodology

- High-level analysis
- Analysis
  - Business requirements
  - Architectural requirements
    - Domain model and Application models
    - Transformation and Functional patterns
  - ProtLIMS: Implementation, Deployment and Testing

Figure 6.0 – MDA Tools and modeling processes

<table>
<thead>
<tr>
<th>Function</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>UML modeling tool</td>
<td>Compuware OptimalJ 3.2</td>
</tr>
<tr>
<td>MDA approach modeling tool</td>
<td>Compuware OptimalJ 3.2</td>
</tr>
<tr>
<td>Source code generation tool</td>
<td>Compuware OptimalJ 3.2</td>
</tr>
<tr>
<td>Java IDE</td>
<td>JetBrains IntelliJ IDEA 4.5</td>
</tr>
<tr>
<td>JSP and HTML code editor</td>
<td>Macromedia DreamWeaver MX 2004</td>
</tr>
<tr>
<td>Database modeling tool</td>
<td>Sybase PowerDesigner 10.0</td>
</tr>
<tr>
<td>Database mining tool</td>
<td>BenthicSoftware Golden 5.7</td>
</tr>
</tbody>
</table>
## MDA approach: ProtLIMS models

<table>
<thead>
<tr>
<th>PIM</th>
<th>DOMAIN MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Domain model describes system independent of implementation technology</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PSM</th>
<th>APPLICATION MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Application model describes system specific to certain technology. Application model for ProtLIMS is based on J2EE technology and contains five submodels: Common, Database, Business logic, Business façade, Presentation</td>
</tr>
</tbody>
</table>

**Database model**
Contains entity-relationship database model

**Common data elements model**
Contains common data elements such as data structures and keys

**Business logic model**
EJB and DAO components describe system business logic

**Business façade model**
Business façade components are used to separate business logic from presentation tier

**Presentation model**
Describes presentation tier in terms of web components and web flows

<table>
<thead>
<tr>
<th>CODE</th>
<th>CODE MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Code model contains all generated source code. System source code contains:</td>
</tr>
</tbody>
</table>

**Database code**
Database scripts for all available databases are generated in code model
Code description: sql scripts

**Common application code**
Contains common data elements such as data structures and keys
Code description: java classes

**Business logic code**
Contains EJB and DAO specific implementations of system business logic
Code description: java classes and configuration files

**Business façade code**
Contains Business façade code for business logic access
Code description: java classes

**Presentation model**
Contains system web code based on struts framework
Code description: java classes, jsp pages, html pages, configuration files etc.
Figure 7.0 – J2EE Patterns Implemented in OptimalJ