PROTEXPRESS 1.0

Technical Guide

NATIONAL® CANCER INSTITUTE

Center for Biomedical Informatics and Information Technology

February 24, 2009

protExpress Development and Management Teams			
Development	***Quality Assurance***	***Documentation***	***Project and Product Management***
Krishna Kanchinadam[2]	Nonna Rabinovitch[5]	Carolyn Kelley Klinger[4]	Carl Schaefer[1]
Scott Miller[2]	Tom Boal[5]	Krishna Kanchinadam[2]	Anand Basu[1]
Steve Matyas[2]		Mahidhar Narra[6]	Xiaopeng Bian[1]
			Bill Mason[2]
			Brent Gendleman[2]
Systems & Application Support			
Bob Wysong[3]	Andrea Johnson[3]	Ralph Rutherford[3]	Sriram Kalyansundaram[3]
Nimish Shah[3]			
1. National Cancer Institute Center for Biomedical Informatics and Information Technology (NCI CBIIT)		2. 5AM Solutions Inc.	3. Terrapin Systems
4. Lockheed Martin Management System Designers	5. NARTech	6. Stelligent	

Contacts and Support	
NCICB Application Support	http://ncicb.nci.nih.gov/NCICB/support Telephone: 301-451-4384 Toll free: 888-478-4423

Contents

About This Guide

This section introduces you to the *protExpress 1.0 Technical Guide*.

Introduction to protExpress

The *protExpress Technical Guide* describes the aspects of protExpress's design that are considered to be architecturally significant; that is, those elements and behaviors that are most fundamental for guiding the construction of protExpress and for understanding it as a whole. Stakeholders who require a technical understanding of protExpress are encouraged to start by reading this document, then reviewing the UML model, and then by reviewing the source code. Please note that all diagrams represented in this document are taken from the protExpress UML model; for more detail about the elements in these diagrams, consult the source model located at: https://gforge.nci.nih.gov/svnroot/gpsxar/trunk/docs/analysis_and_design/models/protexpress.eap.

Purpose

The *protExpress Technical Guide* provides a comprehensive architectural overview of the protExpress system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the system.

Existing protExpress documentation can be found on the protExpress page of the caBIG Tools page located at: http://cabig.nci.nih.gov/tools/protExpress.

Note: Uniform Resource Locators (URLs) are used throughout the document to provide sources for more detail on a subject or product.

Definitions and Acronyms

- **J2EE –** Java 2 Enterprise Edition
- **Java SE –** Java Standard Edition
- **JDK –** Java Development Kit
- **JPA –** Java Persistence API
- **JSP –** JavaServer Pages
- **POJO –** Plain Old Java Object
- **RUP –** Rational Unified Process
- **UML –** Unified Modeling Language

Organization of the Manual

The *protExpress Technical Guide* contains the following chapters:

- Chapter 1, *Overview of protExpress*
- Chapter 2, *Goals and Constraints*
- Chapter 3, *Use Cases*
- Chapter 4, *Architecture*
- Chapter 5, *Implementation Artifacts*
- Chapter 6, *Deployment*

Additional References

For more information about protExpress, see the following references:

- protExpress UML Models
  https://gforge.nci.nih.gov/svnroot/gpsxar/trunk/docs/analysis_and_design/models/protexpress.eap
- protExpress Vision Document
  https://gforge.nci.nih.gov/svnroot/gpsxar/trunk/docs/requirements/protexpress_vision.doc
- protExpress Use Case Summary
  https://gforge.nci.nih.gov/svnroot/gpsxar/trunk/docs/requirements/use_cases/use_case_summary.doc

Text Conventions Used

This section explains conventions used in this guide. The various typefaces represent interface components, keyboard shortcuts, toolbar buttons, dialog box options, and text that you type.

Convention	*Description*	*Example*
Bold	Highlights names of option buttons, check boxes, drop-down menus, menu commands, command buttons, or icons.	Click **Search**.
URL	Indicates a Web address.	http://domain.com
text in SMALL CAPS	Indicates a keyboard shortcut.	Press ENTER.
text in SMALL CAPS + text in SMALL CAPS	Indicates keys that are pressed simultaneously.	Press SHIFT + CTRL.
Italics	Highlights references to other documents, sections, figures, and tables.	See *Figure 4.5*.
Italic boldface monospace type	Represents text that you type.	In the **New Subset** text box, enter ***Proprietary Proteins***.
Note:	Highlights information of particular importance.	**Note:** This concept is used throughout this document.
{ }	Surrounds replaceable items.	Replace {last name, first name} with the Principal Investigator's name.

Chapter 1 Overview

An ever increasing amount of proteomics research data, especially mass spectrometry data, has been made available in the past few years. Several large scale repositories have been created to host proteomics experiment and protocol data. The National Cancer Institute has implemented a public proteomics repository, based on the Computational Proteomics and Analysis System (CPAS), developed at the Fred Hutchinson Cancer Research Center. An instance of the system currently hosted at NCICB serves valuable data from NCI-funded cancer proteomics initiatives to the scientific community. Currently, the repository provides public access to 10,160,229 peptide identifications from 687 ms-ms runs from the Mouse Proteomics Technology initiative (MPTI).

One hurdle to the easy movement of data from labs to public repositories is the lack of standard formats and tools for capturing experiment and protocol annotations. For example, CPAS uses the eXperimental ARchive (XAR) format, while other tools use proprietary and custom formats. While the current NCICB proteomics repository can import XAR data, it lacks the appropriate tools to support the creation of XAR files. Currently, these XAR files are manually created by the data curators, which is a very time consuming and error-prone process.

protExpress, a web-based tool for capturing proteomics experimental annotations, aims to fill this gap. protExpress is an open source project utilizing industry standard best practices that is developed on the J2EE platform. Encompassing a subset of the FuGE object model, it provides a framework for describing experimental procedures and steps. An intuitive web-based interface allows users to input and manage experiment and protocol information, enabling researchers to specify an experiment with a series of protocols, with specific inputs and outputs. The software exports experiment information into the XAR format used by CPAS. The architecture of protExpress allows for the addition of other annotation formats in the future. It also provides a programmatic application programming interface (API) to allow access to the underlying data.

Chapter 2 Goals and Constraints

caBIG Silver Compliance

protExpress must be implemented in such a way that it may be certified caBIG Silver com¬pliant. While Silver compliance is the requirement, protExpress will provide a grid interface in anticipation of a possible move to Gold level compliance when the criteria for Gold compliance are established.

Supporting Multiple Formats

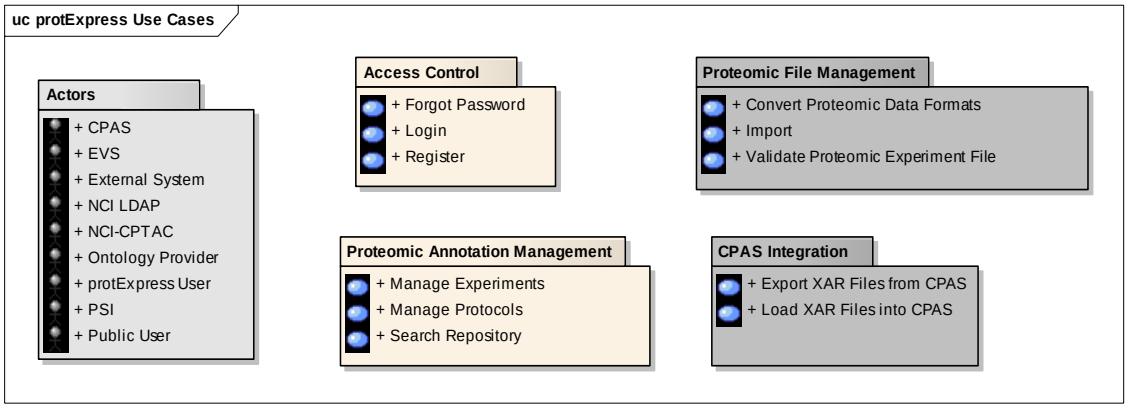
protExpress must be implemented in such a way that it may be possible to support multiple proteomic formats in the future.

Chapter 3 Use Cases

The use cases represented in this section contain the functionality that have the greatest impact on the design of the protExpress architecture. In brief, the use cases described in this chapter require implementation of mechanisms for security, validation, file management, data storage and retrieval, and API design. Brief descriptions of each of these use cases are provided below as extracted from the model.

For information on the complete use-case model see the protExpress Use-Case Summary document located in the SVN repository at the following location: https://gforge.nci.nih.gov/svnroot/protExpress2/trunk/docs/requirements/protExpress_use_case_summary.doc.

An overview of the use cases is depicted in the following diagram.



- Denotes actors in the system.

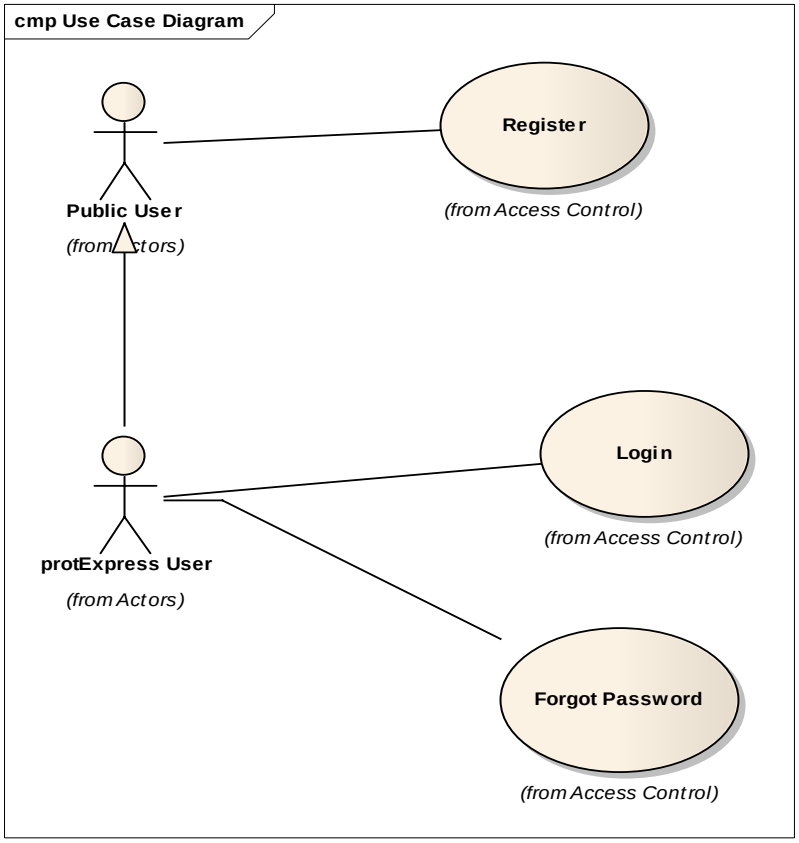
- Denotes the use cases that are implemented in protExpress 1.0.

- Denotes use cases that have been deferred and not implemented in protExpress 1.0.

protExpress Use Cases

Access Control

Initiated by a user, the login use case allows for registration of a non-registered user for a new protExpress account, allowing them to login to access the application.



Register

Initiated by a non-registered user, this use case allows the user to request a new account. This will enable the user to login to the system and perform certain actions for which they have the appropriate privileges.

Login

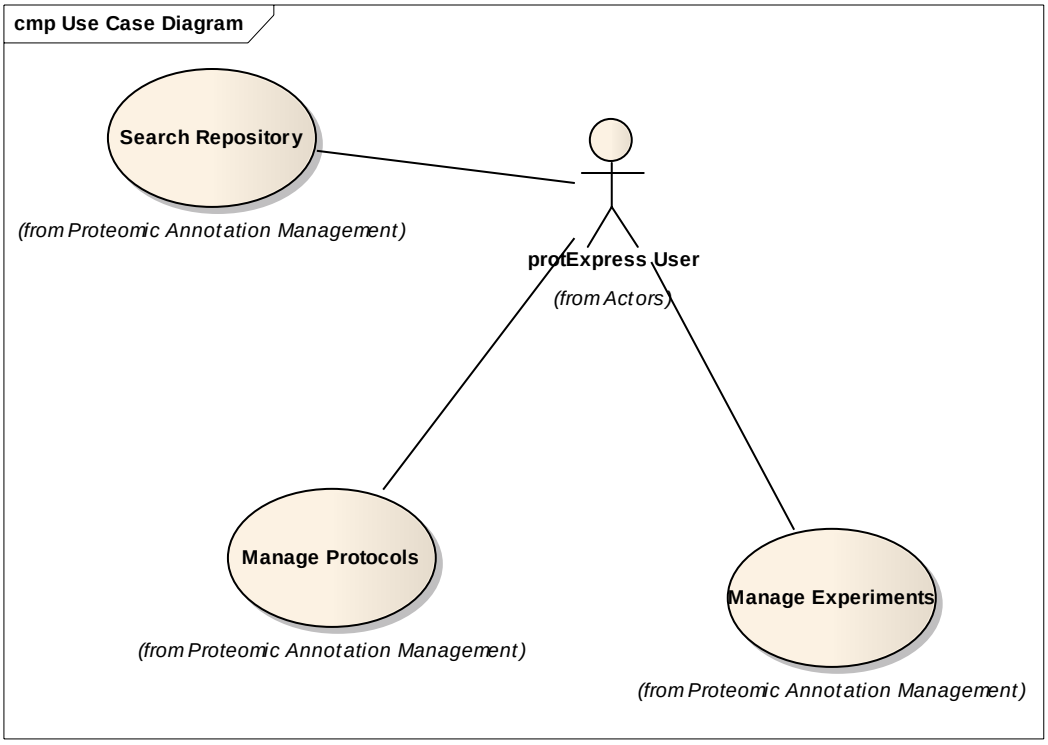
Initiated by registered protExpress user, this use case allows for the validation of the authenticity and authority of the given user either against a networked (LDAP) set of users or a local set (database). As a result of a successful login, the registered user is presented with their home space and the set of operations they have privileges to perform.

Forgot Password

Initiated by registered protExpress user, this use case allows the user to request help from the System Administrator to recover and/or reset their forgotten password, to enable them to re-access the system.

Proteomic Annotation Management

Initiated by a registered and logged in user, this use case enables the input of protocols, experiments, protocol inputs and outputs, and annotation information into the system. The ability to search the data is also provided.



Manage Protocols

Initiated by a protExpress user, this use case allows the user to input and create new protocols, and modify/delete existing protocols from the system.

Manage Experiments

Initiated by a protExpress user, this use case allows the user to input and create new protocols, and modify/delete existing experiments from the system. The following functionalities are supported:

1. Create, modify and delete experiments.
2. Provide ability to repeat an experiment. This will result in multiple “runs” of the experiment, with each run representing an instance of the experiment. Each experiment has an individual set of protocol applications, with each protocol application having its own sets of inputs and outputs.

3. Export experiment data into an eXperimental ARchive (XAR) format used by CPAS.

Note: protExpress architects envision that protExpress will be able to support a variety of formats. However, for version 1.0, only the XAR format is supported.

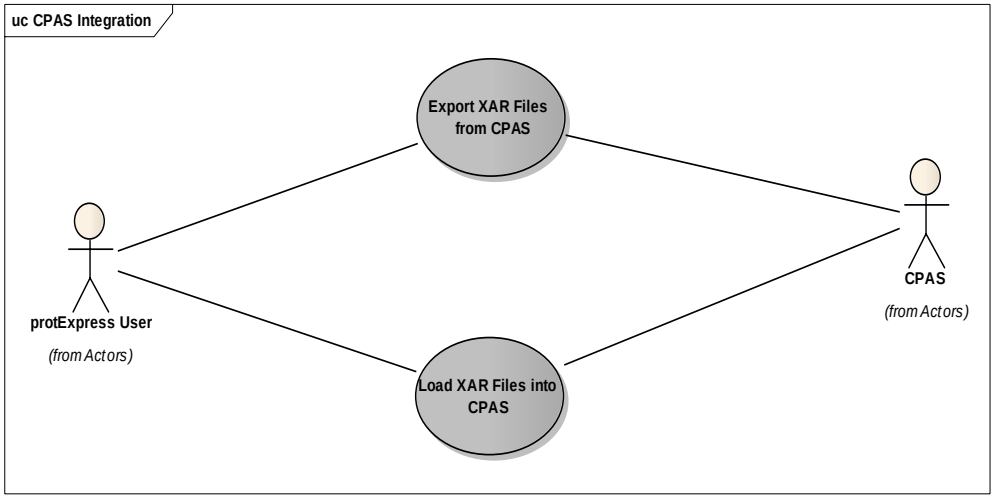
Search Repository

Initiated by a protExpress user, this use case a user of the system to search and view experiments and protocols present in the system. Checks are made to ensure that a user can only edit/delete the entities they have privileges to do so.

CPAS Integration

Initiated by a registered and logged in user, these use cases allow a registered user of the system to communicate with an installation of CPAS (Computational Proteomics Analysis System)

Note: This use case has been deferred and not implemented for protExpress 1.0.



Load XAR files into CPAS

Initiated by a protExpress user, this use case allows the user to export the experiment information into a XAR formatted XML file. This file can be uploaded into a CPAS installation for further processing.

Note: This use case has been deferred and not implemented for protExpress 1.0.

Export XAR files from CPAS

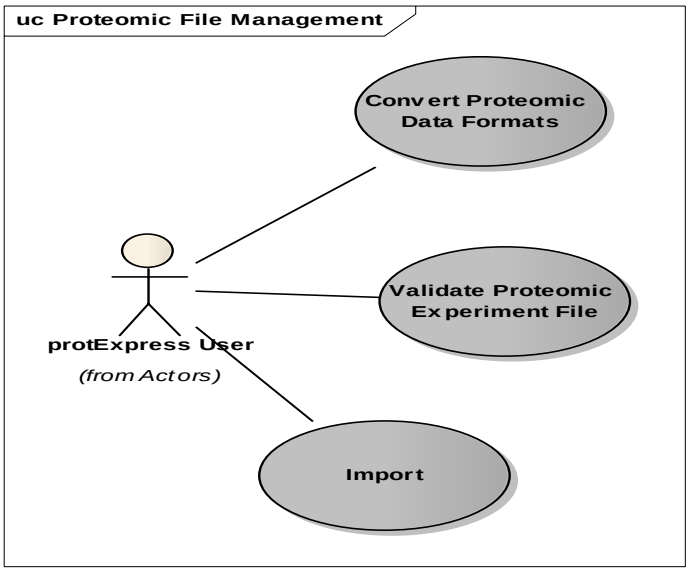
Initiated by a protExpress user, this use case will allow an XAR formatted XML file (exported from CPAS) to be imported into a protExpress system.

Note: This use case has been deferred and not implemented for protExpress 1.0.

Proteomic File Management

Initiated by a registered and logged in user, this use case enables the conversion of experiment data from/into multiple formats supported by the system, in addition to validating the syntax and semantics of the generated format.

Note: This use case has been deferred and was not implemented for protExpress 1.0.



Convert Proteomic Data Formats

Initiated by a protExpress user, this use case will allow the user to retrieve an experiment data from the system and convert it to a supported format.

Note: This use case has been deferred and was not implemented for protExpress 1.0.

Import

Initiated by a protExpress user, this use case will allow the user to import proteoemics experiment data from a supported format into the protExpress system.

Note: This use case has been deferred and was not implemented for protExpress 1.0.

Validate Proteomic Experiment File

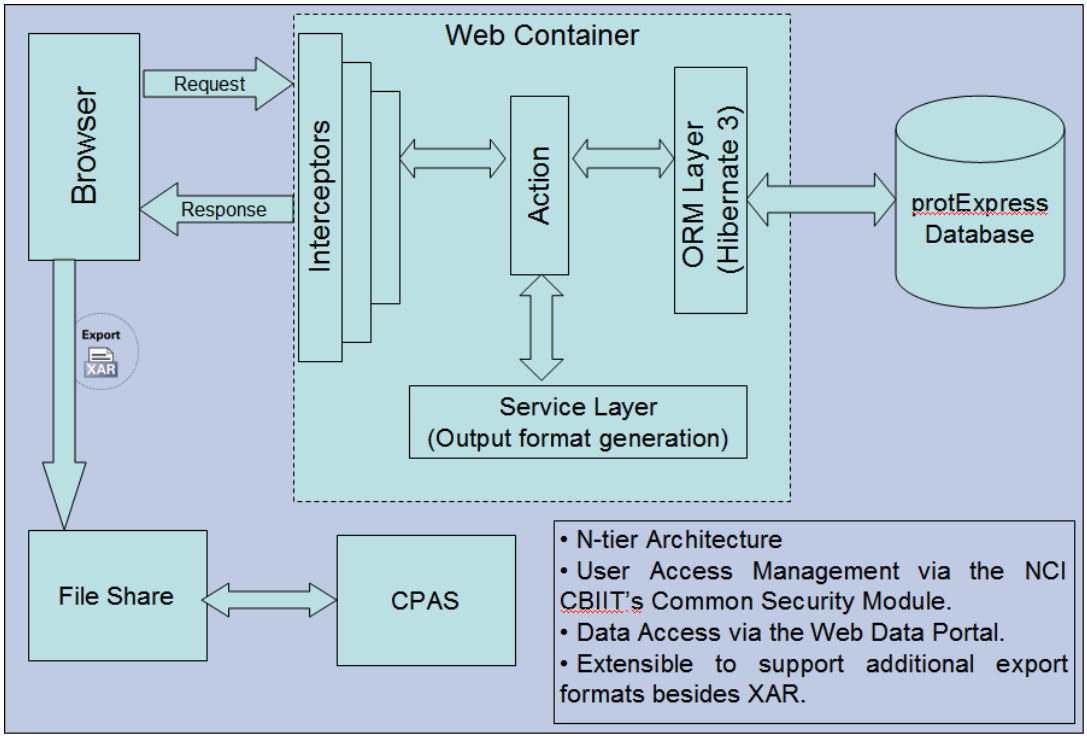
Initiated by a protExpress user, this use case the user to take a file containing proteomic experiment and annotation information, and validate the syntax and semantics of the experiment data using the protExpress system.

Note: This use case has been deferred and was not implemented for protExpress 1.0.

Chapter 4 protExpress Architecture

Overview and Workflow

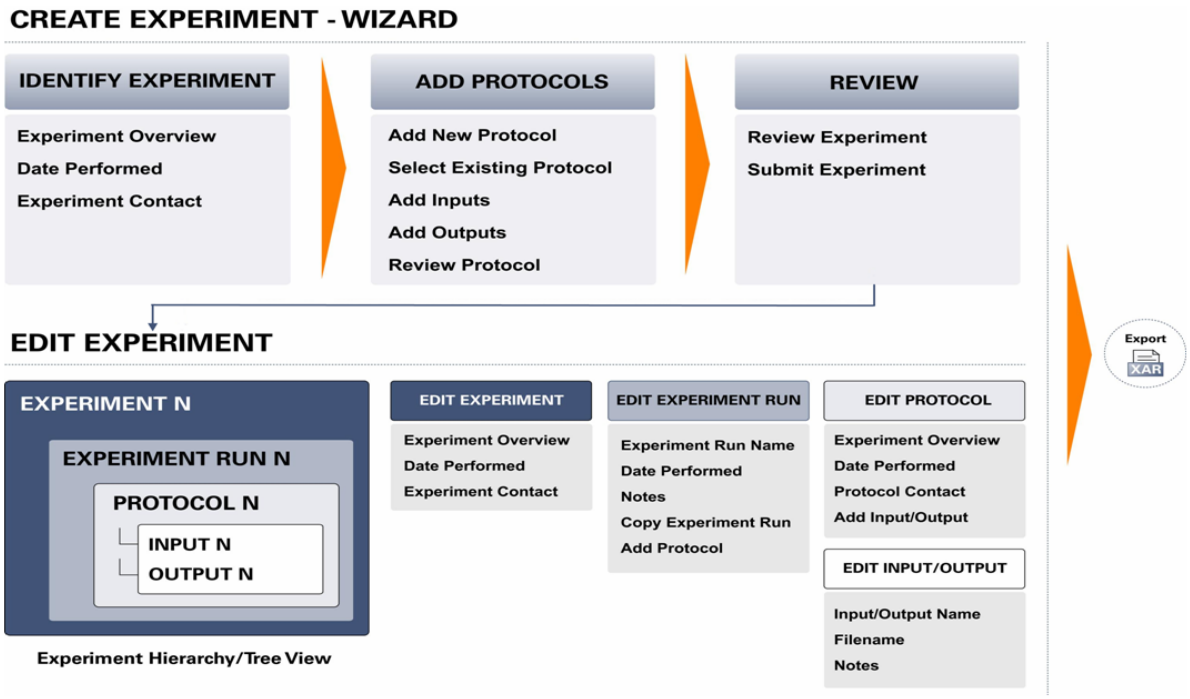
protExpress is implemented as a J2EE 1.5 application built on top of Java SE 5 (JDK version 1.5.0_11) employing core J2EE technologies. Persistence is being managed directly with Hibernate 3.2 rather than Java's persistence API (JPA). JPA provides no significant advantages over Hibernate at this point, and Hibernate provides additional extended functionality not included in JPA. Clients of protExpress can be characterized as either web UI clients or API clients. The basic architecture of the system is depicted via the following diagram:



protExpress 1.0 is implemented as a J2EE web application employing Struts 2 as the Model-View-Controller implementation. The web application is deployed into the Apache Tomcat 5.5.20 container. The system interacts with the underlying database via Hibernate 3 as the ORM layer. User access management is achieved via the NCI CBIIT's Common Security Module.

Experiment and annotation data can be exported from the system as a XAR formatted XML file, that can be uploaded to a common file share accessible by CPAS. Subsequently, the XAR file can be uploaded to CPAS for further processing.

The diagram below depicts the workflow of the system:



More detailed description of the architecturally significant elements is provided in the following next sections.

Architecturally Significant Design Elements

User Interface Layer

The protExpress user interface is accessed as a standard web application via HTTPS. It is implemented as a J2EE web application employing Struts 2 as the Model-View-Controller implementation. This layer provides presentation, navigation and validation functionality only. Validation logic at this level is limited to standard form-based validation (for example, checking for appropriate field formats) and is implemented using Struts 2 validation. Furthermore, a bridge from Struts 2 validation to the Hibernate Validator framework was implemented that allows the definition of these constraints to come straight from the data model. This ensures that the UI enforces the same constraints applied by the underlying storage mechanism. All application logic is implemented in the lower layers of protExpress.

The pages presented to the web client use HTML and JavaScript only. No applets or other client-side component technologies are used. Many pages are dynamically updated based on user input without a complete page refresh using Ajax. This allows us to improve responsiveness, implement tabbed interfaces, and improve application usability. Ajax functionality is provided using the Prototype and Scriptaculous JavaScript libraries, the Ajaxtags tag library, and the Struts 2 Ajax functionality provided by Dojo.

The User Interface layer also includes the login and authentication logic via the integration of NCI CBIIT's Common Security Module (CSM) authentication into the

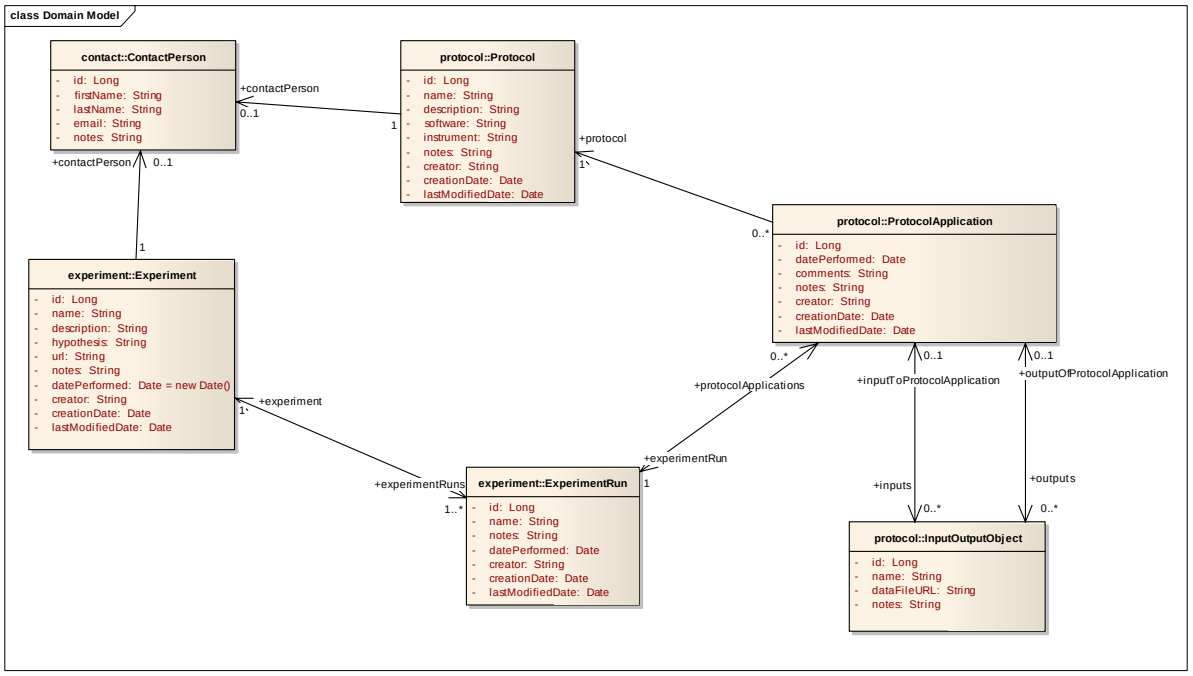
J2EE standard security model, allowing for both database- and LDAP-based authentication.

Grid API

The protExpress Grid API is a Grid 1.2 compliant data service created via the Introduce Toolkit. The service connects, via JNDI and RMI, to a running instance of the protExpress Remote Java API. The grid service connects to the web app at startup and services all requests received from the grid. All data in protExpress is available via the data service.

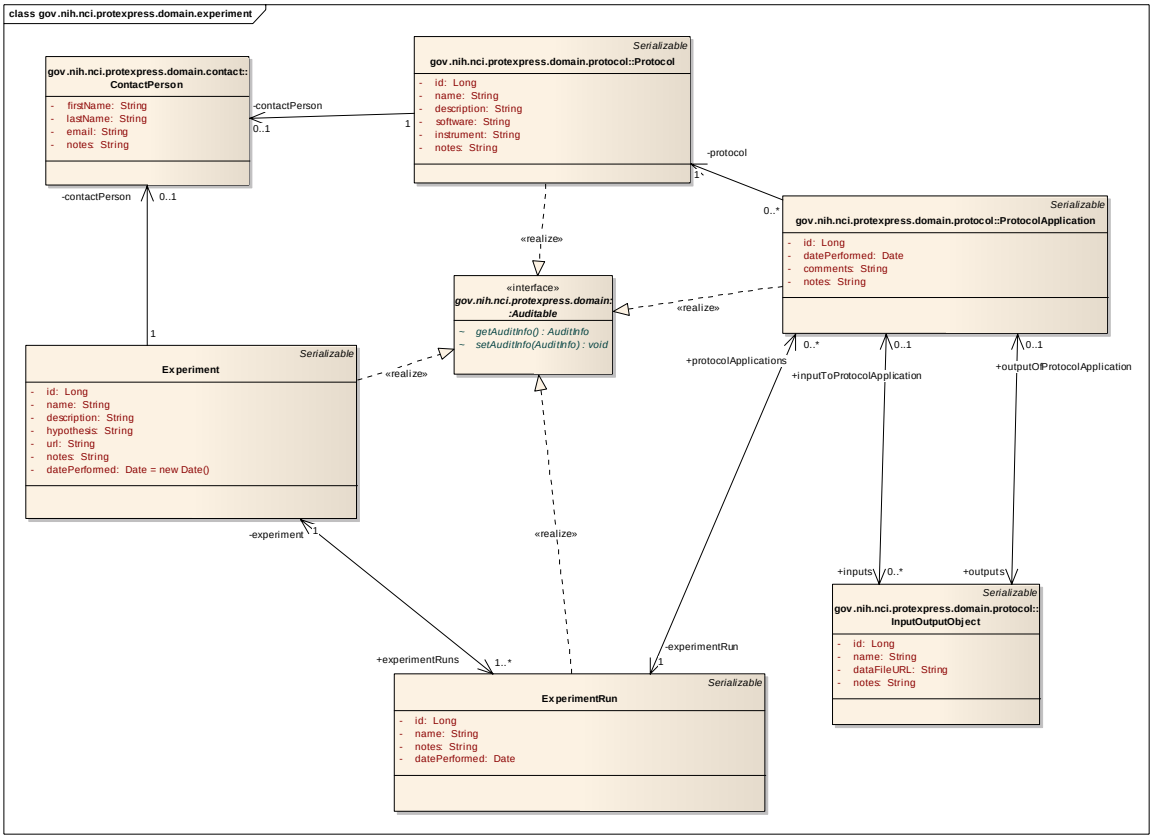
The Grid Transfer framework solves the serialization problem by providing an out of band channel for retrieving the binary data. Instead of returning the data directly and serializing it inside the SOAP response, the data is staged on the server, a WS-RF resource is created for the data, and a reference to this resource is returned to the client. The client then uses this reference to initiate a transfer of the actual data over a separate HTTP connection. If Grid Transfer is used in conjunction with Grid Security, then an HTTPS connection is used and all security credentials held by the client are applied. For more on Grid Transfer, see its page on the caGrid wiki at http://www.cagrid.org/wiki/CaGridTransfer.

The domain model for the grid service generated by the Introduce toolkit is given below:



protExpress Domain Model and Classes

This section describes the domain model and classes used to model the experiment and protocol data that protExpress is designed to manage. The diagram below depicts the domain model for the system:



The underlying business object model is implemented as a set of POJOs that model the domain of proteomics experiment and protocol data. The protExpress implementation encompasses a small subset of the FuGE object model.

Chapter 5 Implementation Artifacts

Overview

The major physical artifacts that comprise the protExpress software deployment units are given below.

Artifacts

protExpress.war

The `protExpress.war` artifact is the J2EE Web Application Archive (WAR) that contains all of the web portal application, that make up the User Interface, Application Logic, and Business Logic layers of the application. It packages the JSPs and Struts 2 classes comprising the User Interface layer of the system, in addition to containing the Struts 2 related and other necessary supporting JAR's.

protExpressGridApp.war

This artifact, generated by the Introduce toolkit, comprises the web application archive that is responsible for querying the experiment and protocol information requested by the grid service.

wsrf.zip

This artifact represents the WSRF resource created by the Introduce toolkit.

Chapter 6 Deployment

The CBIIT deployment of protExpress involves a front-end Apache web server that receives the HTTPS requests and then delegates these requests to the Apache Tomcat 5.5.20 server where protExpress is deployed.

Globus, the grid service execution environment, is deployed within the same Apache Tomcat container 5.5.20. Howerver, external adopters might also choose to deploy application components and execution environments to a single server or multiple servers.

Appendix A Glossary

Terms used in this guide are defined below.

Term	***Definition***
API	Application Programming Interface
protExpress	Protein Express
caBIG	cancer Biomedical Informatics Grid
WAR	Java Web Application Archive
Javadoc	Tool for generating API documentation in HTML format from doc comments in source code (http://java.sun.com/j2se/javadoc/)
JDBC	Java Database Connectivity
JUnit	A simple framework to write repeatable tests (http://junit.sourceforge.net/)
LDAP	Lightweight Directory Access Protocol
ORM	Object Relational Mapping
UI	User Interface
WAR	Web Application Archive
WSDL	Web Services Description Language
XMI	XML Metadata Interchange (http://www.omg.org/technology/documents/formal/xmi.htm) - The main purpose of XMI is to enable easy interchange of metadata between modeling tools (based on the OMG-UML) and metadata repositories (OMG-MOF) in distributed heterogeneous environments
FuGE	Functional Genomics Experiment model.

Index