EU Project KBBE.2011.3.6-01 NewProt

D1.1 SSP Framework
*NewProt Platform Requirements and Design Specification*

Peter Haase (fluid Operations)

Michael Schmidt (fluid Operations)

Andreas Schwarte (fluid Operation)

tbc

**Abstract**

The Self Service Portal (SSP) will provide a platform that will give users access to a broad range of protein engineering tools and services. On an embedded website users will interactively work with these tools and perform their computations supported by integrated workflows. The Self Service Portal will provide a tight integration of standard tools for protein engineering and workflow development.

This document is used to specify requirements as well as system architecture and application design of the SSP.

Reqirements are gathered following follows IEEE 830. The requirement specification include a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the specification also contains non-functional requirements (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

The design specification follows IEEE 1016-1998, also known as the Recommended Practice for Software Design Descriptions, an [IEEE](http://en.wikipedia.org/wiki/IEEE) standard that specifies an organizational structure for a software design description.

Keyword list: requirements, design, self-service portal, protein engineering

WP1 Design and implementation of SSP

Prototype/Report/Other PU/PP/RE/CO

Contractual date of delivery Actual date of delivery

**CHANGES**

N.B. This section to be deleted before submission to the Commission

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Changes** |
| 1.0 | 10.02.12 | Andreas Schwarte, Peter Haase | Initial draft |
| 1.1 | 13.6 |  | Updated sections 3 + 4 |
| 1.2 | 14.8 |  | Use cases |

NewProt Consortium

This document is part of a research project funded by the KBBE Programme of the Commission of the European Communities as project number KBBE.2011.3.6-01.

1. Stichting Katholieke Universiteit,
2. Ernst –Moritz-Arndt –Universität Greifswald (EMAUG),
3. Bio-Prodict B.V. (BIOP),
4. EnzymiIcals AG (ENZYM),
5. Yasara Biosciences GMBH (YAS),
6. Ingenza Limited (INGEN),
7. S.A.F.AN. Bioinformatics - Sas Di Pugliese Luisa (SAFAN),
8. Fluid Operations AG (FLUID),
9. Lead Pharma Holding B.V. (LEAD),
10. Enatis S.R.O. (ENANTIS)

Contents

NewProt Consortium 2

Executive Summary Error! Bookmark not defined.

Contents 2

1 Introduction 2

1.1 Background and Motivation for the SSP 2

1.2 Definitions 2

1.3 Methodology and Structure of the document 2

2 Overall description 2

2.1 Product perspective and Use Cases 2

2.1.1 Existing relevant protein engineering tools 2

2.1.2 Selection of Use Cases for the proposed system 2

2.2 Product functions 2

2.3 User characteristics 2

2.4 Constraints, assumptions and dependencies 2

3 Specific requirements 2

3.1 External interface requirements 2

3.2 Functional requirements 2

3.3 Performance requirements 2

3.4 Design constraints 2

3.5 Logical database requirement 2

3.6 Software System attributes 2

4 SYSTEM ARCHITECTURAL DESIGN 2

4.1 Chosen System Architecture 2

4.2 Discussion of Alternative Designs 2

4.3 System Interface Description 2

5 DETAIL DESCRIPTION OF COMPONENTS 2

5.1 Component n **Error! Bookmark not defined.**

5.2 Component n+1 **Error! Bookmark not defined.**

6 USER INTERFACE DESIGN 2

6.1 Description of the User Interface 2

6.1.1 Screen Image 2

6.1.2 Objects and Actions 2

# Introduction

## Background and Motivation for the SSP

Researchers and practicians working on protein engineering can select from a broad variety of tools and services to address common workflows. For many of these workflows several tools are required with the consequence that intermediate results have to be passed between the different tools with manual effort. Although there exist standards for data formats as well as interfaces for exchange of intermediate results, a tight integration of the different tools supporting practical use-cases is missing.

The Self Service Portal (SSP) will provide a platform that will give users access to a broad range of tools and services. On an embedded website users will interactively work with these tools and perform their computations supported by integrated workflows. The Self Service Portal will provide a tight integration of standard tools for protein engineering and workflow development. Throughout the workflows intermediate results can be retrieved and visualized in a unified user interface.

The SSP will be designed based on the open source Information Workbench from fluidOps. The SSP will provide a working environment to easily interact with all produced NewProt resources. To this end, the SSP will hold tabs for database access (for WP2), software access (for WP3), system facilities (svn access, ftp access, virtual machine, etc), dissemination (e.g. course material, software documentation) help facilities, and project management facilities. Users will be able to interactively run NewProt software within the SSP workbench.

Software and databases will be fully interoperable, i.e., users will not need to store in-between results and will not need to worry about file formats etc. This interoperability requires that all data types will be syntactically and semantically described (WP2) and that all software that can operate on those data will need to 'know' about the syntactic and semantic annotations (WP3-5). The use of common standards (RDF, Linked Data, as well as relevant domain ontologies such as EDAM, OBO and the GO Gene Ontology)) will ensure the semantic interoperability of data and software. Much of the data and software the partners intend to incorporate in the SSP already use the SOAP protocol for interoperability and adhere to these common standards and ontologies.

The SSP will be provided as a hosted portal (hosted at the CMBI), but it will be designed such that all partners can instantiate private copies (e.g. for in-house use) if desired. Two mechanisms will be used for this purpose.

First, all resources (except third party databases) will be kept in the svn version control system, and second, the whole database and software suite will be made available as a Virtual Machine image that can be instantiated

on private infrastructures (e.g. using VMware) or, if needed, on public clouds (e.g. Amazon EC2). Partner FLUID will design the SSP keeping in mind that they will be able to include their template based provisioning tools when, in due time, they deliver technological support to SSP-using industries that request such additions.

## Definitions

## Methodology and Structure of the document

We follow standard guidelines from software engineering for requirements analysis and design specification.

In particular, we create a software requirements specification (SRS) and software design description (SDD).

A [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints). The requirement specification follows IEEE 830.

A software design description (SDD) is a document used to specify system architecture and application design in a software related project. For the Software Design Descriptions we follow IEEE 1016-1998, an [IEEE](http://en.wikipedia.org/wiki/IEEE) standard that specifies an organizational structure for a software design description.

# Overall description

## Product perspective and Use Cases

###  Existing relevant protein engineering tools

The starting point for the Self-Service Portal is a set of existing tools that are used by protein engineers and researchers today, albeit with limited interoperability between them. We provide brief description of the relevant tools and their purpose:

1. Yasara
2. MRS
3. 3DM
4. Hotspot Wizard
5. What If
6. Hope

TODO: HSSP Database (Maarten would be expert on this)

### Selection of Use Cases for the proposed system

In this section, we present a set of selected use-cases. These use-cases will illustrate the possibilities for integration and the benefits for users.

|  |  |
| --- | --- |
| Use case 1 | Match a sequence to protein identifier |
| Use case 2 | Finding PDBs associated with a Protein ID |
| Use case 3 | Linking proteins to alignments |
| Use case 4 | Compute model with Yasara |
| Use case 5 | Use Hotspot Wizard output as input for Yasara |
| Use case 6 | Generate HOPE report for a protein |

****

|  |
| --- |
| Use case 1: Match a sequence to protein identifier |
| Objective | The objective of this use case is to retrieve an existing protein identifier matching the user-provided sequence. Each protein in the system is identified by such an ID and additional information (such as for instance models) is associated to it. Proteins in the Newprot portal will be identified by their Uniprot ID. |
| Stakeholders | This functionality addresses all users of the portal as it provides an entry point for protein lookup.  |
| Significance | With the integration of the search interface into the portal, a single point of access is established. Users do not have to employ multiple different services, nor do they have to deal with data exchange between different tools.  |
| Methods | 1. The user provides a sequence in FASTA format (or plain text) using a text field
2. A protein identifier is retrieved using the Blast Service
3. Result is visualized in the portal
 |
| Components | * MRS Blast Service
* NIH (official Blast) for commercial applications
 |
| Risk | Through the use of external services a dependency on these services is introduced with respect to the quality. This might in particular become a risk for the commercial (closed-source) Blast service from NIH. |

|  |
| --- |
| Use case 2: Finding PDBs associated with a Protein ID |
| Objective | The objective of this use case is to search for or lookup an existing PDB file using the given protein identifier. The protein model is maintained in the local database of the Newprot portal. |
| Stakeholders | This functionality addresses all users of the portal as it provides an entry point to retrieve protein models as PDB file, the basis for many use cases |
| Significance | The protein model represented by a PDB file is the basis for most scenarios in the Newprot system (in particular UC3, UC5 and UC6). This mode is used as input for all protein engineering tools. |
| Methods | 1. If the PDB file is already available in the local database, continue with step 4.
2. Search for PDB file at MRS using the Uniprot ID
3. Store the PDB file in the local database
4. Show a summary page for the given protein, including a link to the PDB file
 |
| Components | * Local database
* MRS
 |
| Risk | Through the use of external services a dependency on these services is introduced with respect to the quality.  |

|  |
| --- |
| Use case 3: Linking proteins to alignments |
| Objective | The objective of this use case is to find the alignments for a protein and to visualize the results in the commercial tool 3DM or in another visualization tool. |
| Stakeholders | Any user of the portal |
| Significance | This use case captures the central link between protein and alignments and provides and easier way to activate operations. |
| Methods | 1. Using the PDB ID as input, the HSSP file is looked up within the MRS HSSP database
2. The retrieved HSSP file is stored in the local database
3. A link to 3DM is created based on the PDB ID

Alternative: If the model is computed (e.g. by Yasara as in UC4), the HSSP needs to be computed as well using a web service.Note: 3DM does not actually need the HSSP files, however, they might be required for other visualization tools. |
| Components | * MRS HSSP database
* 3DM
* Alternative visualization tools (e.g. jalview.org)
* HSSP generation web service
 |
| Risk | A risk might be the identification and the integration of alternative visualization tools |

|  |
| --- |
| Use case 4: Compute model with Yasara |
| Objective | The objective of this use case is to compute a protein model using the Yasara client. This use case captures the scenario where the model cannot be retrieved from an existing database. |
| Stakeholders | Any user of the portal |
| Significance | This use case complements UC3 with the possibility to compute protein models if the PDB file cannot be retrieved using an existing database. |
| Methods | 1. Using the protein sequence, Yasara or the Protein Model Portal are used to compute the PDB file
2. The PDB file is stored in the local database
3. Show a summary page for the given protein, including a link to the PDB file

Discussion: use Yasara Client or Server for computation? |
| Components | * Yasara (client or server)
* PMP service (= Protein Model Portal)
 |
| Risk | Computation of protein models from the sequence is a lengthy process. Risks are the computational costs as well as concurrency issues (i.e., parallel computation, management of parallel computation tasks, etc.) |

|  |
| --- |
| Use case 5: Use Hotspot Wizard output as input for Yasara |
| Objective | The objective of this use case is to use the hot spots computed with the Hotspot Wizard as additional input for Yasara. In Yasara these hot spots can be used for further analyses. One goal of this use case is to facilitate protein engineering workflows in terms of data exchange between standard tools, i.e., the intermediate results are passed between the tools automatically. |
| Stakeholders | Any user of the portal |
| Significance | The tight integration of the Hotspot Wizard and Yasara from a user’s perspective, support and simplify standard protein engineering workflows as intermediate results do not have to be passed manually between the tools |
| Methods | 1. Using a PDB file the Hotspot Wizard computations are invoked
2. The Hotspot Wizard (asynchronously) returns the hotspots in the Newprot Format (NPF)
3. User selects NPF file to trigger visualization in Yasara

Note: on a job page the user can track running or past computation jobsQuestion: What does Yasara actually generate as visualization |
| Components | * Hotspot Wizard
* Yasara
 |
| Risk | A risk might the computation costs (in particular CPU power) as well as concurrency issues. In addition, an agreement on the data interchange format – the Newprot Format (NPF) – has to be found. |

|  |
| --- |
| Use case 6: Generate HOPE report for a protein |
| Objective | The objective of this use case is to generate a HOPE report for a given protein. An example scenario could be the question of which mutation to make for a given protein to achieve stabilization. |
| Stakeholders | Any user of the portal |
| Significance | The integration of this use case into the portal provides a single point of access and simplifies user workflows |
| Methods | 1. Using a PDB file (c.f., UC2/UC4), HOPE is used to build a homology model
2. HOPE results are embedded in the portal accociated with the protein (Polling for results)
 |
| Components | * HOPE
 |
| Risk |  |

**NPF Format:**

Requires format for representing residues (Mapping of Residue to Residue Value, plus Header)

Stakeholders for format: Yasara, 3DM, Hope, Hotspotwizard (coordinated by fluidOps)

Use case ordering:

1) UC5

2) UC1

3) UC2

4) UC4

5) UC3

6) UC6

Use case on getting from a sequence to a HSSP file (UC7)

- recognize that PDB does not exist

- request an HSSP file from HSSP web service (Maarten)

**Use case on inferencing**

Infer useful user actions based on previous actions of other users

## Product functions

## User characteristics

## Constraints, assumptions and dependencies

**What do you mean by: Using Hope for complex visualizations**

First visualization: Residues from HotSpotWizard as input for Hope

Security aspects: ACLs within shared portal or private instances of the portal?

# Specific requirements

## External interface requirements

The SSP must provide the following external interfaces:

* End-users can access and use the SSP via a Web Browser, using either HTTP or HTTPS protocols. The system should support the latest versions of the most prominent browsers (in particular the latest versions of Internet Explorer and Firefox). The HTML pages delivered by the SSP are plain HTML5/JS pages, such that the basic functionality can be accesses without installing any external software or browser plugins (e.g., no Flash Player needs to be installed). Note, however, that this restriction does not necessarily hold for external components that are integrated into the SSP.
* Access to data residing inside the SSP is possible through a SPARQL interface, following the official *W3C SPARQL Protocol* for RDF recommendation. Using this protocol, authorized users can extract and export data residing inside the system into several standard serialization formats such as RDF/XML, N3, JSON, etc.
* Administrators can access the SSP via a Command Line Interface (CLI), using remote method invocation (RMI) for communicating with the backend. The CLI provides basic functionality for administrating the system, including system backups and the manipulation of data residing inside the system.

## Functional requirements

**F1:** User registration process

The user must be able to register to the system. What can a non-registered user do at all?

**F2:** Upload and delete PDB Files

**F3**: Communication with MRS

* Search in MRS database from inside the SSP
* Transfer/Interlink PDB files related to MRS search inside the system (similar to F2)

**F4**: PDB file and residue library

* Survey page listing available PDB file, which can be chosen for further processing
* Same for residues (which, for instance, could have been generated from HotSpot Wizard)
* Links between PDB files and residues (“provenance” information)

**F5**: Communication with Yasara Server:

* Trigger scripts on Yasara server running in the client
* Option 1: Yasara delivers png, must be possible to display inside the SSP
* Option 2: Yasara delivers scene file, which can be displayed with a Yasara client running on the user’s machine (see F6)

**F6**: Communication with Yasara Client:

* Feed a Yasara client running on the user’s computer with a PDB file from the SSP
* Feed a Yasara client with a scene file from the SSP

**F7**: Communication with 3DM

* Link proteins, amino acids, and alignments to pages in the corresponding 3DM system.

**F8**: Communication with HotSpotWizard

* Process PDB file from SSP’s PDB library with HotSpot wizard (integrated into the SSP)
* Store the residue delivered by the HotSpot Wizard inside the SSP

**F9**: Workflow management from the portal

* Selection of inputs (two files)
* Use external tool to compute output
* Get some value from the generated output
* Display this value (associated to the workflow)
* Email notifications

**F10**: Workflow requiring external computational resources

* Selection of inputs (one file, one folder – Mio.)
* Use external tool to compute output
	+ All combinations have to be computed in a loop
* Get some specific value from each generated output
* Take top-k combination w.r.t the output value
* Use top-k combinations for further (long computation) mutations => 1month
* Present final outcome
* Email notifications

## Performance requirements

* We define the response time as the interval between a user-command and the receipt of feedback from the system. In the context of interaction with a Web-based system like the SSP, this means the interval between the user action (such as a click on a link or button) and the delivery of an initial page including feedback (yet not necessarily the final result). The response time of the SSP’s HTML interface shall be less than 2 seconds in 90% of the cases, and less than 10 seconds in 99% of the user requests.

However, this requirement is relaxed to hold for the main navigation pages of the SSP only. For survey pages constructed based on cost-intensive computing steps (such as the aggregation of data from different sources) or external tools embedded into the SSP higher response times can be tolerated.

* The SSP shall scale to at least 20 simultaneous users. This means that the response time requirements above shall hold for 20 simultaneous users. Up to 50 simultaneous users, an increase of the number of simultaneous users shall lead to an at most proportional increase in the response time limits defined above.
* The SSP shall have an availability of 99.9% (not including announced down-times of at most 3 days per year for maintenance).
* The SSP shall have an automatic fault recovery mechanism, which recovers a consistent state of the system in case of a critical error.

## Design constraints

This section describes the technology and design constraints for the SSP.

* Standard Compliance
	+ The HTML delivered by the SSP shall be HTML5-conform
	+ The access to the SPARQL interface shall be conform with the W3C’s latest SPARQL protocol specification
	+ The SSP must be designed in such a way that it is fully functional using the latest versions of the most important Web browsers.
* Hardware Constraints
	+ The SSP requires at least 8GB RAM and a quad core 2.00GHz CPU (or comparable hardware)
	+ The SSP shall be installable on Amazon AWS cloud infrastructure as well as private clouds driven by VMware.
* Software Constraints
	+ The SSP shall be implemented as a platform-independent Java application.
	+ The SSP is designed to run on 64bit systems.
* Other Constraints
	+ The SSP requires Internet connection to communicate with the external protein engineering tools.

## Logical database requirement

*This should specify the logical requirements for any information that is to be placed in a database. This may include the following:*

* *a) Types of information used by various functions;*
* *b) Frequency of use;*
* *c) Accessing capabilities*
* *d)* ***Data entities and their relationships***
* *e) Integrity constraints*
* *f) Data retention requirements*

## Software System attributes

### Security

* Use of common security standards (HTTPS, SSH, …)
* Logs & audit trails
* User authentication mechanism (RSA-certified, if needed?!)
* Protection against DoS-attacks?!

### Availability

* Support for active-passive failover configuration
* Load balancing support
* Automatic recovery, backups

### Maintainability

* Developed on standards
* CLI for maintenance at runtime

### Portability

* Implemented in Java
* Avoidance platform-dependent libraries
* Use of standardized web-technologies

### Testability

* Reached through modularization & code design

### Usability

* …

# SYSTEM ARCHITECTURAL DESIGN

In this section, we present the data model and the architecture of the SSP. The portal will be designed based on the open source Information Workbench from fluidOps and will provide a working environment to easily interact with all produced NewProt resources. A design goal of the SSP is to allow for project based collaboration between users.

In the remainder of this section we present the data model of the portal, discuss the chosen architecture, present a set of alternative design options, and finally close with a description of the system interface.

## Data Model

A design goal of the SSP is to provide a unified view on the NewProt resources, including computational tools, databases and interfaces. Users will be able to interactively run NewProt software within the SSP workbench, while all Software and databases are fully interoperable, i.e., users do not need to store in-between results and do not need to worry about file formats etc. To achieve this level of interoperability between different computational tools and services, the data model will be based on standard formats, e.g. PDB signatures for proteins. The overall data model is depicted in Figure 1 and described in detail in the following.

In the portal data is organized in a way that allows collaborative workflows involving multiple users. These workflows are represented by the notion of projects which have a number of associated users - the project owners or contributors. Note in particular that multiple users can work on the same project. Within these projects all associated resources for the project are maintained, i.e. computational workflows and past results.

Projects provide a unified view on a user’s ongoing work: contributors to the project can see all relevant information about the project, i.e. the current state, all applied resources, and the computational results. In a project one or more of the NewProt Software components are used for performing computations on the proteins defined by workflows. In the data model this is reflected by the 1:N relation of projects to proteins, i.e. one project can use multiple proteins as resources.

To allow for interoperability between the NewProt Software components, proteins are represented with standard formats. In particular, this is the PDB signature of the protein and the HSSP definition. These representations are maintained in files which are associated to the particular protein.

For visualization of proteins we associate scenes to the PDB representation of a protein. This is either a rendered PNG or a description of the scene in SCW format which can be processed further by other NewProt components. These visualizations are also maintained in the file system and can be displayed in the portal on-demand (associated to a protein).



Figure 1: Overview of the data model

Protein: Sequence as attribute

HOPE reports as external documents with own metadata in the portal

Files could be local files or external documents, generally

Proteins: Will be uniquely identified in the scope of a project, linking to a UniprotID/URI

Job: owned by a user, linked to a protein, from there to a project

## Chosen System Architecture

The Self Service Portal integrates multiple computational tools as well as services relevant for protein engineering. As already discussed, the major design goal of the SSP is to provide the user with a unified view on the NewProt resources. In the following, we describe the chosen architecture for the SSP and discuss the integration of these components.

### The Information Workbench

The architecture of the SSP is designed based on the open source edition of the Information Workbench from fluidOps. Serving as a platform for Linked Data, the Information Workbench enables users for collaboration, integration of public as well as private data and services, and analytics on the data. Users benefit from a unified view both on the data level and on the integrated resources.

TODO maybe one or two paragraphs on the Information Workbench architecture?

TODO running as a web application => J2EE

In the NewProt project the Information Workbench is the layer of integration, providing a unified way of accessing data and components. Components in this sense are the standard tools for protein engineering provided by the project partners. A detailed description about the individual components is provided in section 5. Below we describe how these components are integrated into the SSP.

### MRS

MRS is one of the NewProt resources to be integrated into the SSP. MRS is a search engine for biological and medical databanks and provides fast access to over a terabyte of indexed data from all standard databanks such as Uniprot or Genbank. In the SSP the MRS interface will be integrated as means to search for proteins. The integration of MRS will on the one hand be done by providing an interface for search in an Iframe, while structured data exchange will be done via RESTful web services. With such integration the user does not have to leave the SSP platform, and has still all resources and services at hand.

As mentioned the MRS search engine will be a facility to search for proteins. In particular it will be used as a source for PDB representations of proteins, which in the SSP serve as input for other NewProt components to perform computations. The PDB files are retrieved from the MRS services via RESTful services and then stored in the local filesystem of the SSP for further use.

### Hotspot Wizard

The Hotspot Wizard is a computational service to automatically compute so-called “hot spots” in protein structures. These “hot spots” can be used for the processes in protein engineering, as well as for annotation of protein structures. As output the Hotspot Wizard provides lists of annotated residues including estimations for mutability.

In the SSP the Hotspot Wizard will be integrated as a web application that runs on the SSP host. From the portal, users can invoke computations on proteins using the Wizard by passing PDB files to the Hotspot Wizard service. Once the computations are finished, the results (i.e. the annotated residues) are

### Yasara

YASARA is a molecular modeling and simulation program including facilities for visualization of such resources. In the SSP, we employ YASARA on the one hand as a rendering engine for protein structures (running as a local server application), and on the other hand as a client application to visualize prepared YASARA scene descriptions.

### 3DM

3DM is a commercial tool with extended capabilities. 3DM provides protein superfamilies systems based on structural superpositions and multiple sequence alignments. The SSP will link alignments, proteins, and amino acids to corresponding entries in 3DM. If the commercial features are activated in the portal a high-quality 3DM system for the selected protein will become available.

### HOPE

Computes results asynchronously, need to poll for results

### Other tools and services

- Hope

- What If

### Extensibility



TODO: Integrate HOPE (in first instance running at CMBI)

## Discussion of Alternative Designs

- tight integration vs. Loosely integrations

- remote webapps vs. local web apps

## System Interface Description

- web based interface (in the Information Workbench)

- API access via CLI / REST

- handling of tools either local, as web app or remote web services

# DETAIL DESCRIPTION OF COMPONENTS

## MRS

MRS provides services that allow searching for proteins and structures using large well-defined databases. In the portal we employ these services to retrieve the PDB signatures of proteins, the input for various other tools. We will embed the elaborate search interface of MRS in the Portal using IFrames. In addition, we will use standard web service techniques (i.e. SOAP or RESTFUL endpoints) for structured data exchange.

## 3DM

3DM is a commercial tool with extended capabilities. The 3DM platform can be used to create knowledgebases for protein superfamilies. Structure based multiple sequence alignments are used to create the alignments. Data can be extrapolated from one protein to another and from one amino acid to another based on the structurally conserved core of the family. 3DM also includes mutator: a text mining tool to retrieve mutations from literature, comulator to calculate correlated mutation analyses, and validator to asses the pathogenicity of unknown variants in proteins. HTML links to corresponding 3DM resources are provided for all proteins, amino acids, and alignments in the portal.

## Hotspot Wizard

The Hotspot Wizard is a computational tool which amongst other features takes PDB files as input and computes relevant residues that can be used by other tools for further analyses and computations. The Hotspot Wizard will be running as a web application local to the Self Service Portal.

## Yasara

For Yasara we distinguish between the Yasara server and the Yasara client. The Yasara server will be running as a server process on the Self Service Portal’s host, while the Yasara client is installed on the user’s machine.

The purpose of the Yasara server is to provide means for rendering PNG scenes of selected residues. The residues of interest are passed to local scripts which create Yasara compatible input data. Using these inputs Yasara produces PNG scenes and stores them in the file system. Since both the Yasara server and the Self Service Portal run on the same host, the rendered PNG scene is immediately accessible from the portal for visualization.

In contrast, the Yasara client is running on the user’s machine providing means for more extensive visualization and further local analyses. From the portal, a local Yasara client is started and the selected SCN object is passed as input.

# USER INTERFACE DESIGN

## Description of the User Interface

### Screen Image

### Objects and Actions

- in year 1 we only have one instance of the portal

Login screen

separate user mechanism

logo on the left page

acknowledgements (EU flag)

User page

- list of projects, create new project (new instance of the portal?)

- current jobs / job status

- search field (also for google/wikipedia)

- ihop as a resource for search

Project page

- running jobs

- associated proteins, add proteins

- associated files

- invite user / add person to project

- entry points for starting workflows

 - get to protein from sequence (UC1), add to project

Protein page

- visualize information from

 - UniProt

 - associated files (visualizations, NPF files, alignments as HSSP, PDB)

- associate / lookup / upload PDB files for Protein (UC2) (+ possibility to add annotations for users)

- trigger actions

 - Compute model with Yasara (UC4)

- Compute HOPE report (UC6)

File pages (general)

- file metadata and provenance

PDB Files

 - Compute HotSpots (UC5)

 - Compute HOPE report (UC6)

 - Visualization of alignments (UC3)

HSSP Files

- Visualization of alignments (UC3 with alternate to 3DM)

NPF Files (Residue information)

- Visualization of HotSpot (UC5)

Job Page

- metadata about status, start time

- files that have been produced

APPENDIX

A. Example PDB signature: Caffeine

ATOM 1 C MOL 1 -1.597 0.768 6.992 inf inf

ATOM 2 N MOL 1 -1.586 -0.581 6.688 inf inf

ATOM 3 C1 MOL 1 -2.832 -1.358 6.759 inf inf

ATOM 4 C2 MOL 1 -0.407 -1.203 6.316 inf inf

ATOM 5 C3 MOL 1 0.779 -0.464 6.248 inf inf

ATOM 6 C4 MOL 1 0.757 0.902 6.559 inf inf

ATOM 7 N1 MOL 1 -0.430 1.506 6.928 inf inf

ATOM 8 C5 MOL 1 -0.451 2.939 7.254 inf inf

ATOM 9 N2 MOL 1 2.003 1.377 6.421 inf inf

ATOM 10 C6 MOL 1 2.823 0.390 6.038 inf inf

ATOM 11 N3 MOL 1 2.087 -0.774 5.921 inf inf

ATOM 12 C7 MOL 1 2.597 -2.095 5.525 inf inf

ATOM 13 O MOL 1 -0.403 -2.510 6.022 inf inf

ATOM 14 O1 MOL 1 -2.744 1.362 7.352 inf inf

ATOM 15 H2 MOL 1 -2.946 -1.770 7.759 inf inf

ATOM 16 H3 MOL 1 -2.795 -2.169 6.035 inf inf

ATOM 17 H4 MOL 1 -3.677 -0.710 6.535 inf inf

ATOM 18 H5 MOL 1 -0.650 3.514 6.352 inf inf

ATOM 19 H6 MOL 1 0.513 3.232 7.666 inf inf

ATOM 20 H7 MOL 1 -1.232 3.132 7.986 inf inf

ATOM 21 H1 MOL 1 3.882 0.487 5.851 inf inf

ATOM 22 H8 MOL 1 2.529 -2.200 4.444 inf inf

ATOM 23 H9 MOL 1 2.003 -2.871 6.003 inf inf

ATOM 24 H10 MOL 1 3.636 -2.190 5.834 inf inf

TER

Bibliography and references

This part should contain the list of documents and other key references relevant to the deliverable.