

Bettina HALWACHS

Master Thesis

**Model Driven Software
Development with an
Application to the Management
of Mutein Data**



Institute for Genomics and Bioinformatics,
Graz University of Technology
Petersgasse 14, 8010 Graz, Austria

Supervisor and Evaluator
Dr. Gerhard Thallinger

Graz, May 2010

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz,
(date)

.....
(signature)

for my parents

Abstract

German

Proteine sind essentielle Elemente in metabolische Prozessen und zellulären Signalkaskaden. Wird die Abfolge der Aminosäuren eines Wildtyp Proteins verändert entsteht ein so genanntes Mutein, welches sich in Struktur und folglich in Selektivität und Funktionalität von seinem Wildtyp unterscheidet. Dies bildet die Grundlage für die Entwicklung von Muteinen, welche individuell für bestimmte katalytische Reaktionen geschaffen werden. Aufgrund des immer größer werdenden Interesses an *maßgeschneiderten* Enzymen im industriellen und medizinischen Bereich, bedarf es einer Applikation, welche es erlaubt Informationen über Muteineigenschaften zu sammeln, zu verwalten und aufzubereiten.

Im Zuge dieser Arbeit wurde unter den Aspekten der Modellgetriebenen Softwareentwicklung eine dreischichtige Java Enterprise Webapplikation entwickelt, welche es ermöglicht die spezifische Aktivität von Enzymen bei der Umsetzung eines bestimmten Substrats in ein oder mehrere Produkte zu erfassen, zu analysieren, zu verwalten und zu präsentieren. Verschiedene Suchmechanismen erlauben es, gezielt nach bestimmten Enzymen und den damit verbundenen katalytischen Reaktionen zu suchen. Aus den Ergebnissen lassen sich einfach jene Enzyme extrahieren, welche ein bestimmtes Substrat am besten katalysieren.

Die dreischichtige Applikation setzt sich aus einer Oracle Datenbank in der Datenhaltungsschicht und dem Anwendungsserver JBoss in der Logikschicht zusammen. Die Präsentationsschicht, die Benutzeroberfläche basiert auf JavaServer Faces, AJAX und JBoss Seam.

Stichwörter: Protein, Mutein, Katalyse, Drei-Schichten-Architektur, MDA, JEE

Abstract

English

Proteins are essential elements in metabolic processes and cellular signalling cascades. Modifications of their constituting amino acid sequences can have serious effects on structure and hence on selectivity and overall functionality of an enzyme. Proteins with changes in their sequential amino acid order are called muteins. In recent years the interest of the medical science community and biotechnology companies in tailor-made muteins for particular catalytic reactions increased dramatically. Therefore, an application was needed that enables storage, maintenance and retrieval of mutein related properties.

In the course of this thesis a three tiered Java Enterprise web application for the management of mutein data was developed based on the model driven software development paradigm. The application provides features for data collecting, maintenance, analysis and presentation of specific activities related to protein as well as mutein data related to the catalysis of a particular substrate into at least one product. Different search mechanisms simplify retrieval specific of enzymes and their related catalyses. Enzymes which catalyse a reaction best can be easily extracted from the search result.

The three tiered Java Enterprise application comprises an Oracle database at the back end and a JBoss application server at the business layer. Business logic is encapsulated in EJB3 and the web interface is built on JavaServer Faces, AJAX and JBoss Seam.

Keywords: Protein, Mutein, Catalysis, Three-Tier-Architecture, MDA, JEE

Contents

List Of Figures	v
List Of Tables	vi
Listings	vii
Glossary	viii
1 Introduction	1
1.1 Thesis Objectives	1
2 Background	3
2.1 Proteins	3
2.2 Muteins	5
2.2.1 Mutation Types	6
2.3 Enzymatic Reactions	7
2.4 Related Databases	9
2.4.1 BRENDA	9
2.4.2 Protein Mutant Database	9
2.4.3 ProTherm	9
2.4.4 SPROUTS	10
2.4.5 SuperCYP	10
2.5 MuteinDB 1.0	10
3 Methods	11
3.1 Architectural Concepts	11
3.1.1 Unified Modeling Language	11
3.1.2 Model Driven Architecture	11
3.2 Technologies	12
3.2.1 Java Enterprise Edition	12
3.2.2 Enterprise Java Beans	13
3.2.3 Java Server Faces	13
3.2.4 JBoss Seam	14
3.3 Tools	14
3.3.1 AndroMDA	14
3.3.2 AndroMDA3 Template Project	15
3.3.3 Entrez eUtils	16
3.3.4 PubChem - Power User Gateway	16

3.3.5	CrossRef Meta Data Service	17
3.3.6	Eclipse	18
3.3.7	MagicDraw UML	18
3.3.8	Maven	18
3.3.9	Oracle SQL Developer	19
3.4	General Nomenclatures	19
3.4.1	Canonicalization	19
3.4.2	CAS Number	19
3.4.3	EC Number	20
3.4.4	Simplified Molecular Input Line Entry Specification	20
3.5	Libraries	24
3.5.1	Apache POI	24
3.5.2	Genome Usermanagement	25
3.5.3	Chemical Development Kit	25
3.5.4	JDOM	26
3.5.5	overLIB	27
3.5.6	JMEMolecularEditor [©]	27
4	Results	30
4.1	Database Structure	30
4.2	Development of UML Diagram	31
4.3	Business Logic	32
4.3.1	Data Import Service	32
4.3.2	Structure Search Service	37
4.3.3	Search Service	38
4.4	Web Application	39
4.4.1	Application Look-And-Feel	39
4.4.2	Search Interface	40
4.4.3	Search Result Presentation	43
4.4.4	Mutein Data Presentation	44
4.4.5	Edit Interface	47
4.4.6	Data Import Interface	49
5	Discussion	51
5.1	Software Technologies	51
5.2	Application Features	52
5.3	Outlook	53
	Bibliography	59
	A JME Licence Agreement	60
	B PUG conversation example	61
	C Usermanagement integration example	64

D	UML Models	65
D.1	MuteinDB 1.0 UML Diagram	65
D.2	MuteinDB 2.0 UML Diagram	66
E	Meta Data Retrieval	67
E.1	XML Response for EFetch GenBankID request	67
E.2	XML Response for CAS 2 CID Transformation	68
E.3	XML Respons for PMID meta data query	69
E.4	CrossRef Meta Data Service Response	70
F	Data Import MS Excel Template	72
G	Data Import Guidelines	73

List of Figures

2.1	Illustration of central dogma of molecular biology	3
2.2	Illustration of the AA Codon-Sun	6
2.3	Activation energy diagram of an enzymatic reaction	8
2.4	Enzym-substrate-complex building	8
3.1	Model Driven Architecture principal concept	12
3.2	Illustration of JEE application model	13
3.3	Model-View-Controller design pattern	14
3.4	Query, Display Setting Function of Report Bean	16
3.5	Illustration of CAS-Number structure	20
3.6	Canonical labelling for generating unique SMILES	24
3.7	Unique SMILES generation example.	24
3.8	JME Editor	28
4.1	Snippet of MuteinDB data import MS Excel file	33
4.2	Main validation phases within the data import	34
4.3	Cell value validation methods	35
4.4	Work flow of substructure searching	38
4.5	General Application layout	40
4.6	Basic search parameter selecting schema.	40
4.7	Search for muteins by wildtype protein	41
4.8	Search for mutein by inhibitor molecule	41
4.9	Search for muteins universal search interface	42
4.10	Search for muteins by reaction parameters.	42
4.11	Substructure search interface	42
4.12	Illustration of default search result presentation	44
4.13	Illustration of mutein specific activity search result presentation	44
4.14	Illustration of expanded mutein specific activity presentation.	45
4.15	Illustration of substructure search result interface	45
4.16	Illustration of basic data tab of the detailed enzyme view.	45
4.17	Illustration of the properties tab within the detailed enzyme view.	45
4.18	Illustration of inhibition tab within the detailed enzyme view.	46
4.19	Sequence tab of detailed enzyme view for mutein sequence data.	46
4.20	Illustration of the substrate tab within the detailed enzyme view.	47
4.21	Illustration of basic data edit view	48
4.22	Illustration of properties edit view	48
4.23	Illustration of inhibition edit view	48
4.24	Illustration of catalysis edit view.	49
4.25	Illustration of the activity edit view.	49

4.26	Example of a positive data import feedback.	50
4.27	Example of a negative data import feedback.	50
D.1	Database schema of MuteinDB 1.0	65
D.2	Database schema MuteinDB 2.0	66
F.1	Illustration of columns that form the MS Excel import file	72

List of Tables

2.1	Amino acids in alphabetical order with their standard abbreviations .	4
3.1	Six classes of the enzymatic classification system	21
3.2	Examples for atom SMILES representation.	21
3.3	Examples for atom SMILES representation with formal charge.	22
3.4	Examples for bond generation rules of SMILES representation.	22
3.5	Multiple SMILES representing the structure of 6-hydroxy-1,4-hexadiene.	22
3.6	Examples of branch representation in SMILES.	23
3.7	Aromatic compound example within SMILES	23

Listings

3.1	Code snippet <code>UniversalIsomorphismTester</code>	26
3.2	HTML snippet for integrating the JME editor into the web front end.	27
4.1	eUtils request URL for sequence data of GenBankId P08684 (CYP3A4)	36
4.2	eUtils URL for transforming CAS (103-79-7) into CID	36
4.3	eUtils URL for querying meta data from PubChem	37
4.4	CrossRef Meta Data Service URL	37
B.1	XML request for retrieving SMILES representation for 3 given CIDs .	61
B.2	PUG XML response containing the request ID	62
B.3	Programmatic poll request to PUG	62
B.4	PUG XML response containing the result	63
C.1	Code snippet for <code>aas:permission</code> user management tag	64
E.1	eUtils XML response for GenBankID P08684	67
E.2	eUtils URL for transforming CAS (103-79-7) into CID	68
E.3	ESummary DocSum response for PMID 10191269	69
E.4	CMS response for DOI 10.1002/adsc.200505069	70

Glossary

AA	Amino Acid
AAS	Authentication and Authorisation System
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
BRENDA	BRaunschweig ENzyme DATabase
CAS	Chemical Abstract Service
CDK	Chemistry Development Kit
CGI	Common Gateway Interface
CID	PubChem Compound ID
CIM	Computant Independent Model
CMS	CrossRef Meta Data Service
CORBA	Common Object Request Broker Architecture
CRUD	Create Read Update Delete
CYP	Cytochrome P450
DB	Database
DD	Data Dictionary
DNA	Deoxyribonucleic Acid
DocSum	Document Summary
DOI	Digital Object Identifier
EC-Number	Enzyme Commission Number
EIS	Enterprise Information System
EJB	Enterprise Java Beans
EU	European Union

eUtils Entrez programming Utils
FK Foreign Key
GFP Green Fluorescent Protein
HIS-Tag Polyhistidine-Tag
HSSF Horrible Spread Sheet Format
I/O Input/Output
IDF International DOI Foundation
IGB Institute for Genomics and Bioinformatics
IMBT Institute of Molecular Biotechnology
IUPAC International Union of Pure and Applied Chemistry
JEE Java Enterprise Edition
JME Java Molecule Editor
JMX Java Management Extensions
JS JavaScript
JSF Java Server Faces
JSP JavaServer Pages
MCS Maximum Common Substructure
MDA Model Driven Architecture
MOF MetaObject Facility
MVC Model View Controller
NCBI National Center for Biotechnology Information
NIH National Institute of Health
OMG Object Management Group
PIM Platform Independent Model
PL/SQL Procedural Language/SQL
PMD Protein Mutant Database
PMID PubMedID
ProTherm The Thermodynamic Database for Protein and Mutants
PSM Platform Specific Model

PTM Post Translational Modification
PUG Power User Gateway
RA Registration Agency
RGraph Resolution Graph
RNA Ribonucleic Acid
RNG Random Number Generator
SMILES Simplified Molecular Input Line Entry Specification
SPROUTS Structural Prediction for Protein Folding Utility System
SQL Structured Query Language
TUG Graz University of Technology
UML Unified Modeling Language
WLN Wiswesser Line Notation
XML Extensible Markup Language

Chapter 1

Introduction

The majority of proteins are enzymes. Enzymes catalyse various chemical reactions. The catalysis of a particular substrate into a special product plays an important role in regulating biotechnological processes. The three major fields which are interested in the understanding and optimisation of catalytic processes are *medicine* and *pharmacy* as well as *industry* and applied *science*. The interests of these groups differ in various ways. While scientists are interested in exploring reasons for the specific activity of an enzymatic reaction, and its role in signalling cascades or metabolic pathways, industry focuses on producing a particular substance (e.g. a plasticiser) in an efficient way. But the most interesting group for society might be the health care aspect. Research results have revealed, that medication depends on patients cell metabolism. Hence, type and dose of a particular drug depend on the patients enzyme variant and has to be chosen individually.

Protein engineering primary deals with the development of tailor-made and useful enzyme variants. Hence, collecting data about enzyme variants could help researchers to get a better understanding of the fundamental properties of enzyme selectivity, activity, stability and structure. Thus, protein design principles get clearer and improve further research approaches.

The information about proteins of interest and their mutations is spread over a wide range of literature sources and databases. Additionally, a single literature source mostly describes a few mutations and their effects without comparing it to other already known mutations. A challenge for protein engineering research groups is the standardised collection, comparison, presentation and accessibility of the available data. A web-based management application for the protein data and their modifications can help to collect, analyse, present and maintain all the information of interest.

1.1 Thesis Objectives

The main objective of this thesis was to develop a management application for mutein data which improves and extends the existing MuteinDB 1.0 features as well as the implementation of new mechanisms using model driven software engineering properties. A redesign of the database schema was necessary because not all of the needed relations were mapped in the existing one. To make use of Web 2.0 technologies and

to adhere to the common application style of the Institute for Genomic and Bioinformatics, it was derived from the AndroMDA3 Template Project [1].

Before starting implementation and research work the following specific thesis goals were defined:

- To use model driven aspects of software development and Web 2.0 technologies within the implementation.
- To redesign and extend the existing database schema according to needs of the research group from the Institute of Molecular Biotechnology at Technical University Graz as well as to needs of planned features.
- To derive MuteinDB 2.0 from the IGB AndroMDA3 Template Project, to benefit from the provided basic mechanisms and technologies.
- To integrate existing features of MuteinDB 1.0 into the new derived application.
- To extend the query mechanism to support the search for enzymes by their wildtype or interacting inhibitor molecules, and also to improve search result presentation.
- To provide a substructure search option
- To implement an automatic enzyme data import mechanism based on data collected in MS Excel files.

Chapter 2

Background

2.1 Proteins

Proteins are essential elements in metabolic processes and cellular signalling cascades. A protein is an organic compound formed by a linear sequence of amino acids (AA). There exist twenty different amino acids (see table 2.1) which are used in protein compositions. Table 2.1 lists these AAs in relation with the correlated one- and three letter notation. The order of this linear chain is encoded in the genome. The central dogma of molecular biology, see fig. 2.1, describes the building of proteins out of deoxyribonucleic acid (DNA) via transcription to ribonucleic acid (RNA) and translation to AA sequence [2].

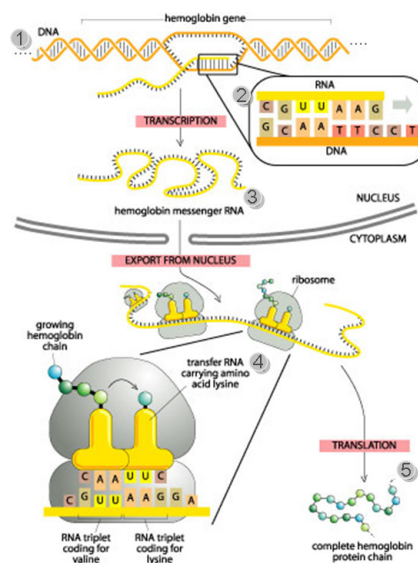


Figure 2.1: Illustration of the central dogma of molecular biology [4]. Within the cell nucleolus: (1) DNA is formed by two single strands that build a double helix. (2) Desaturated section of the DNA, with focus on the transcribed RNA strand (yellow) from the DNA pattern. (3) Finished transcribed RNA strand which is released into cytoplasm. Within cytoplasm: (4) RNA docks to the ribosomes where the protein synthesis takes place. During Translation proteins are formed by nucleic base triplets of the RNA strand. (5) Complete protein chain.

DNA is included in each cell of an organism. It stores the whole genetic information. Every human cell contains the same genes in its nucleus, but different cell

Amino Acid Name	3-letter code	1-letter code
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamic acid	Glu	E
Glutamine	Gln	Q
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Table 2.1: The table represents the twenty AAs in alphabetical order with their standard one- and three letter notation [3].

lines use different genes for their protein synthesis. Cells know which genes they have to activate to build the proteins they need for their specific functionality. The three dimensional structure of DNA is formed as double-helix, see fig. 2.1 (1), where each strand is a composition of the four nucleic bases¹. During transcription, (the conversion from DNA to RNA), DNA strands (orange) are separated, to enable the creation of the RNA strand (yellow) (2). RNA is very similar to DNA, it is also made of a strand of four nucleic bases, but instead of thymine, uracil (U) is used. After transcription the RNA strand (3) is released from the nucleolus to the cytoplasm. There it docks to the ribosomes (4) where protein synthesis takes place. During translation (the generation of proteins from RNA), amino acids are encoded by nucleic base triplets of the RNA strand. After the process has finished the complete protein chain is free available in cytoplasm (5). According to the chemical composition at the end of the protein chain the end is termed N-, C-terminal end respectively. The end of the

¹Adenine (A), Guanine (G), Cytosine (C) and Thymine (T)

AA sequence which is terminated by an AA with a *free amine group* (NH_2) residual is called the N-terminal end. Respectively the end terminated by an AA with a *free carboxyl group* ($COOH$) residual is named the C-terminal end.

The vast amount of proteins are enzymes which catalyse chemical reactions. Thereby they play import roles in metabolic processes like digestion and even the regulation of the transcription of DNA.

The features of enzymes primary depend on the constitution of their *catalytic centre*. Hence, geometric structure and physicochemical properties of this region define selectivity (which substrates are able to bind to the enzyme) and consequently functionality of the enzyme.

Wildtype Proteins

A protein sequence which occur naturally in an organism or the first occurrence of an artificial protein sequence is termed the wildtype (protein).

2.2 Muteins

If the amino acid sequence of a wildtype protein is modified, a new variant is created - the so called **Mutein**. Mutations can lead to incorrectly built proteins which are degenerated in their functionality.

These modification can be either caused naturally, due to environmental influences like radiation or deliberately in the lab. All of these modifications are done on the nucleotide level in different ways see sec. 2.2.1.

For a better understanding of why these modifications can seriously effect cell functionality it is necessary to look at how proteins are encoded by DNA.

A single amino acid is encoded by a nucleic base triple, a so called codon. This results in 64 (4^3) possible codons for twenty amino acids (fig. 2.2). Using the codon-sun in fig. 2.2 the sequence is build beginning from the centre moving to the margin, building direction five prime (5') to three prime (3') end. Direction within a DNA/RNA strand is caused by the different terminal end groups of a strand. The five prime end (5') is terminated by a phosphate group whereas the three prime end (3') has a terminal hydroxyl group. During DNA synthesis the DNA polymerase is just able to add new nucleic bases to ends with a terminal hydroxyl group. As a consequence *building direction* results to 5' -> 3'.

Obvious from fig. 2.2 there are multiple codons encoding for a single amino acid - the genetic code is redundant. For example the codons GCT, GCC, GCA, GCG all code for alanine (A). Because of this redundancies it is possible that different DNA sequences represent the same protein.

Note: Protein synthesis is always initiated by the methionine code (ATG), the so called *start codon*, and stopped by one of the three possible *stop codons* - TAG, TAA, TGA.

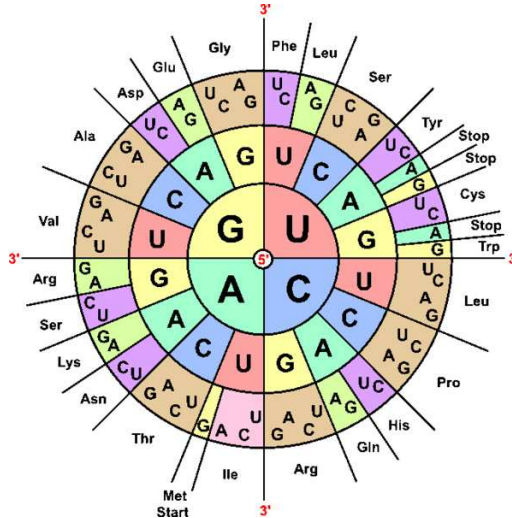


Figure 2.2: Illustration of the AA Codon-Sun [5]. It represents the 64 possibilities for protein encoding out of the four nucleic bases. The sequence is build from the centre moving to the margin (5' -> 3'). For example the codons GCT, GCC, GCA, GCG all code for alanine (A).

2.2.1 Mutation Types

Due to the modification of the AA sequence, the protein can change in structure, molecular weight and length (except in case of substitution). These changes are responsible for the difference in the mutein and the wildtype protein. Especially structural changes are responsible for effecting the catalytic centre of the enzyme. It can become more difficult or even impossible for the substrate to bind. Furthermore the enzyme can become selective to new substrates and as a result able to catalyse other reactions.

The following enumeration gives an overview about mutation types which are considered at the amino acid level.

- Insertion - single or multiple AA(s) are added
- Deletion - single or multiple AA(s) are removed
- Substitution of a single AA
- Truncation
- Fusion
- Tag-Modification

The modification of a single amino acid of the sequence caused by insertion, deletion or substitution is called *point mutation*. In the course of change also a set of multiple amino acids can be inserted or deleted from the wildtype sequence. Whereas insertions, deletions and substitutes occur within the AA sequence, truncations, fusions and tag-modifications arise at the N- or C-terminal end of the sequence.

Truncation

Truncation terms the removal of a significant subset of the sequence at the N- or C-Terminal end of the sequence.

Fusion

The addition of a whole enzyme to another enzyme at the N- or C-terminal end of the sequence is called fusion.

Tag Modifications

There is a wide range of different tags which can be added to the sequence. For example the polyhistidine-tag (HIS-tag), a simple sequence motif mostly formed by six histidine residues. The number of the histidine residues which form the tag can vary. Adding a HIS-tag to an enzyme is a way to pull down proteins for purification. Another widely used tag is the green fluorescent protein-tag (GFP-tag). In contrast to the short HIS-tag the GFP-tag is a whole protein which exhibits green fluorescence when excited with light of a certain wavelength (520 nm). This feature is used for localisation of a particular protein within a cell. All tags related to tag modifications, can be attached to the enzyme sequence at the N- or C-terminal end of an enzyme only.

2.3 Enzymatic Reactions

Biochemical reactions can be accelerated via the use of a biocatalyst. The used enzyme (biocatalyst) catalyses the reaction by decreasing the activation energy². The diagram in figure 2.3 illustrates the activation energy of a reaction with and without a biocatalyst.

The activation energy is significantly decreased by an enzyme, whereas the overall energy which is released during the reaction remains the same.

Other molecules can interact in the enzymatic reaction. The activity of the reaction can be decreased or even suppressed due to an inhibitor molecule. Activators, like co-proteins, are able to support the catalysis which increases the specific activity of the reaction.

The reaction starts with binding the substrate to the active centre of the enzyme - the *substrate-enzyme-complex* (see fig. 2.4) is built. The catalysis of the substrate to the reaction product(s) is then enabled by the enzyme. All the catalysed products are released after the reaction has finished.

Enzymes are known for their high selectivity to the substrate. All enzymatic reactions adhere to the *lock and key model* [7]. This high selectivity is caused by the geometric shapes and physicochemical properties of enzyme and substrate which have to be complementary for building the substrate-enzyme-complex.

²Energy that must be overcome in order to start a chemical reaction.

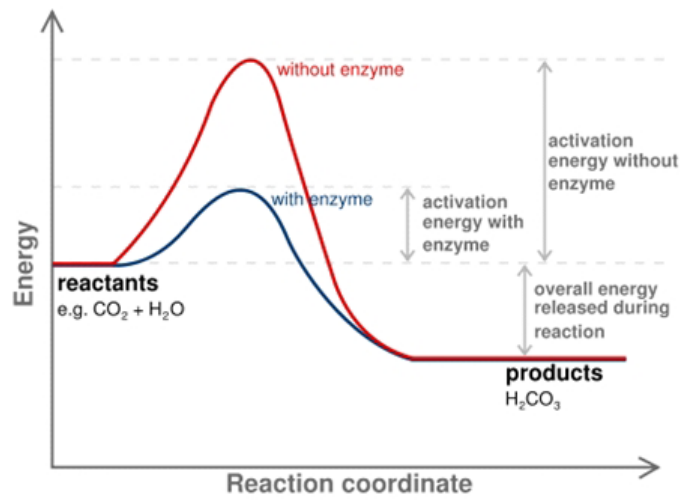


Figure 2.3: Activation energy of a reaction with and without a biocatalyst [6]. The comparison shows with and without the use of an enzyme a stable energy over reaction time. In contrast activation energy decreases significantly with the use of an enzyme than without.

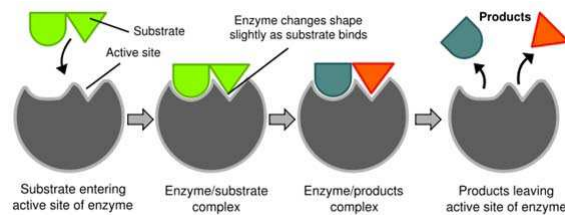


Figure 2.4: Illustration of the enzyme-substrate-complex building [8]. From right to left: The enzyme (grey) with its active centre waiting for the inverse formed substrate (green) to bind. Enzyme and substrate formed the enzyme substrate complex. During catalysis the it is transformed into the enzyme product complex. Finally the substrate has been catalysed, and two products (blue and read) are released.

2.4 Related Databases

Protein-mutation data is handled by few web applications only. Some of them only contain information about a specific enzyme like SuperCYP for Cytochrome P450 data, and none of them provide information about mutain kinetics or the option to search for mutains by their reaction data.

2.4.1 BRENDA

BRENDA³ (BRaunschweig ENzyme DAtabase) - The Comprehensive Enzyme Information System [9], a development of the Department for Bioinformatics, Technical University Braunschweig is one of the most known and acknowledged enzyme databases worldwide. This information system mainly contains biochemical and molecular biological data about enzymes and metabolic pathways as well as a small section about protein mutations [9].

2.4.2 Protein Mutant Database

The Protein Mutant Database⁴ (PMD) developed by the Center for Information Biology and DNA Data Bank of Japan [10], contains artificial and natural mutations of almost all proteins. The database contains information about more than 81.000 mutants from about 10.000 articles. The features provided by the PMD are mentioned below.

- generation, representation of the mutain sequence out of the given information.
- 3D structure display, if experimentally determined 3D structure information is available.
- sequence homology search
- summarised display of mutations of a homologous sequence

2.4.3 ProTherm

ProTherm⁵ the Thermodynamic Database for Protein and Mutants, a project of the Kyushu Institute of Technology, Japan [11] focuses on the collection of thermodynamic parameters which have been obtained from denaturation experiments. These thermodynamic information is important to gain insights in the structure and stability of proteins. The data is registered for wildtype proteins as well as for mutains. Additionally information about secondary structure, experimental design and the used methods for the activity information measurement are available. ProTherm provides search mechanisms with different sorting options for data representation and a 3D structure display option in which mutation sites are mapped automatically [11].

³<http://www.brenda-enzymes.org/>

⁴<http://pmd.ddbj.nig.ac.jp/>

⁵<http://gibk26.bse.kyutech.ac.jp/jouhou/jouhoubank.html>

2.4.4 SPROUTS

SPROUTS⁶ (Structural Prediction for Protein Folding Utility System) is a database for the evaluation of protein stability upon point mutation [12]. The application focuses on the visualisation of protein folding characteristics and the analysis of how point mutations effect protein structure [13].

2.4.5 SuperCYP

"*SuperCYP is a comprehensive resource focused on CYPs and drug metabolism*" [14]. Data collection in SuperCYP⁷ is restricted to the Cytochrome P450 (CYP) enzyme. The database contains information about the enzyme's isoforms and about known mutant isoforms and about their interactions with drugs. The application provides a download option for homology-molded structures in PDB format, data analysis functionality and a result representation option with PubMedID relations [14].

2.5 MuteinDB 1.0

A first prototype of the *MuteinDB*⁸ was developed in 2005. It was implemented by Vincent Rabin based on the user requirements compiled by Michael Guggemos [15]. This version is currently in use by the research group of the Institute of Moleculare Biotechnology⁹, Technical University of Graz¹⁰.

⁶<http://bioinformatics.eas.asu.edu/springs/Sprouts/projectsSprouts.html>

⁷<http://bioinformatics.charite.de/supercyp/index.php?site=home>

⁸<https://muteindb.genome.tugraz.at/>

⁹<http://www.imbt.tugraz.at/>

¹⁰<http://www.tugraz.at/>

Chapter 3

Methods

3.1 Architectural Concepts

3.1.1 Unified Modeling Language

The Unified Modeling Language (UML) [16] is a standard defined and managed by the Object Management Group¹¹ (OMG) for graphical software engineering.

It can be used to specify application structure, functionality, behaviour, architecture, as well as business process and data structure. Modelling application architecture also means structuring the collection of self-contained methods and components of the application/software. Structure can be seen as a way of dealing with complexity. A structured software design enables scalability, maintainability, increases security and guaranties robust execution [17].

UML adheres the MetaObject Facility¹² (MOF) paradigm which is a core concept of Model Driven Architecture, see sec. 3.1.2.

3.1.2 Model Driven Architecture

Model Driven Architecture (MDA) [18] is a model driven software development approach introduced by the OMG. The main concept of MDA is the separation of technical aspects from design. The MDA approach comprises four main models which all describe the same software systems in four different ways. Figure 3.1 illustrates the flow from the Computation Independent Model¹³ (CIM) to the Code Model.

In general UML is used to generate Platform Independent Models¹⁴ (PIM). Based on this system description the MDA development tool starts with the translation of the PIM into a Platform Specific Model¹⁵ (PSM). This PSM is used for creating a running implementation on middle wear platforms like JAVA, .Net or CORBA.

¹¹<http://www.omg.org/>

¹²<http://www.omg.org/mof/>

¹³Textual description of the business model with focus on the user requirements.

¹⁴A model of a software system describing functionality and behaviour, which is independent from programming language or platform. Furthermore it does not include any technical aspects.

¹⁵Model of a software system describing functionality and behaviour using technical aspects.

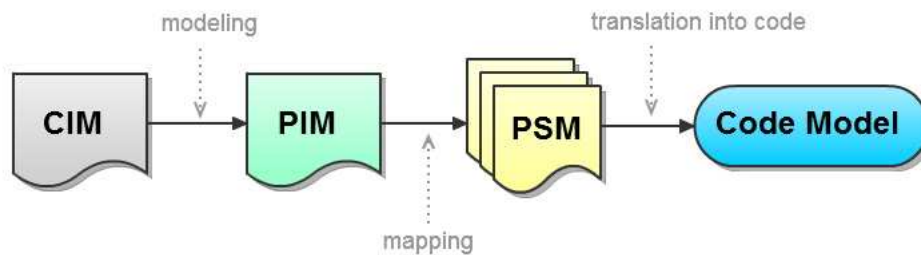


Figure 3.1: Illustration of the principal concept of Model Driven Architecture. Based on the textual description in form of a Computational Independent Model the Platform Specific Model is created using an Unified Modeling Language modelling tool. Using MDA tools the PIM is translated into a PSM which is finally translated into executable code.

Figure 3.1 illustrates the four basic steps of MDA. The main benefits of the MDA approach are listed below.

- focus on business functionality and behaviour
- flexibility, portability towards new standards
- improvement of maintainability
- decrease of programming effort
- reduction of errors
- improvement of testing and simulation facilities

3.2 Technologies

3.2.1 Java Enterprise Edition

Java Platform, Enterprise Edition - JEE 5 [19], is a software specification defined for the architecture of distributed, transaction based and portable web applications. The primary goal of JEE is to simplify writing distributed business applications by providing ready-made enterprise-class services on which developers can rely while focusing on the business logic.

Figure 3.2 illustrates the architecture defined by the JEE application model. Multitiered applications provide flexible services which are easy to access and maintain. The model comprises four main components which form the three different tiers.

- **Client Tier** The client tier comprises all applications running on the client machine like browser or applets.
- **Web Tier** Also called presentation tier. It is responsible for rendering and presentation of the data. For example a JavaServer Faces (JSF) or JavaServer Pages (JSP) page.
- **Business Tier** Contains the business logic of the application.
- **EIS** The Enterprise Information System represents the database at the back end, which stores and provides the data.

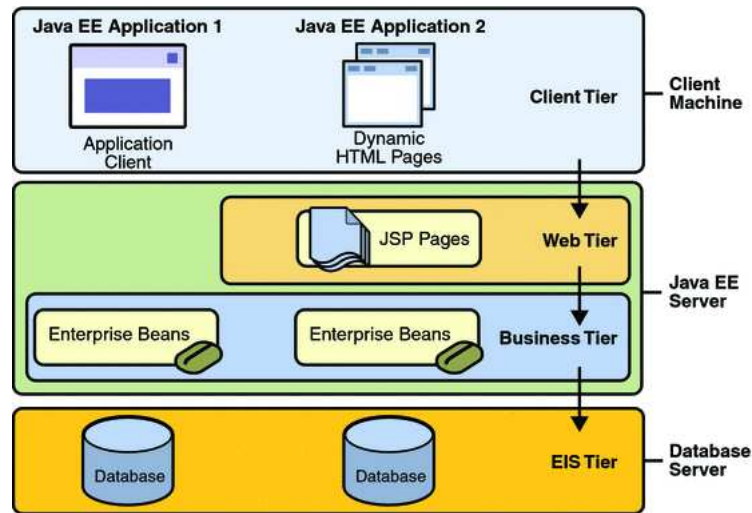


Figure 3.2: Illustration of JEE application model [20]. It comprises three main tiers. The first tier, the persistence tier (EIS), at the back end is responsible for data storage. The second tier, business tier, contains executable code for fulfilling the purpose of the application. Finally, the presentation tier displays the data and handles events.

3.2.2 Enterprise Java Beans

Enterprise Java Beans 3 -EJB3 [21] is one of the core features of the JEE 5 platform. It is a server-side component that encapsulates all the code that fulfils the purpose of an application. The two main goals of EJB3 are simplifying the development of Enterprise Java applications and the creation of a standardised Java Persistence API.

3.2.3 Java Server Faces

Java Server Faces (JSF) [22] is a standardised server side framework which simplifies the presentation tier development of web applications. It belongs to the core frameworks of JEE 5.

JSF adheres the Model-View-Controller (MVC) design pattern. The MVC design pattern, illustrated in fig. 3.3, comprises three components.

The Model The main task of the model consists of storing and providing the data. Mostly the data is managed by a JavaBean.

The View The view is responsible for rendering the data from the model. It knows about the model- as well as about the controller interface. **Note:** There is no logic within the *view*, only display functionality is provided.

The Controller The controller represents the control system of the application. It take care about page navigation and sends messages to the *view* if the *model* has changed. All the logic which is necessary for running the program is contained in the *controller*.

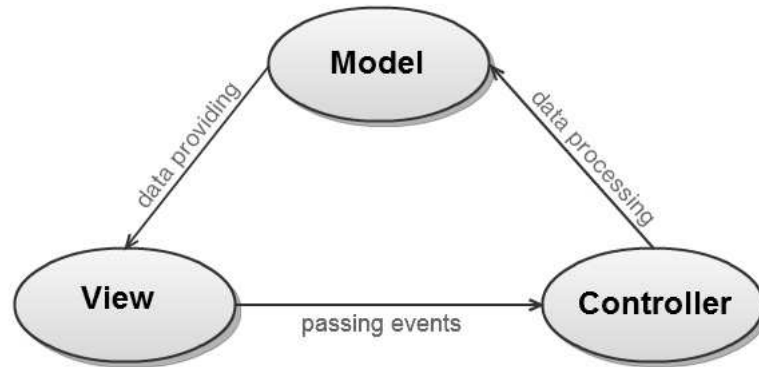


Figure 3.3: Visualisation of the model - view - controller interaction. The data is stored and provided by the model. This data is rendered by the view which passes events to the controller. Controller is responsible for processing the data and passing the modified values to the model.

The use of the MVC design pattern allows to create flexible web applications where presentation tier and business logic is strictly separated. This decoupling leads to better maintainability and scalability of the web application.

JBoss Rich Faces

RichFaces [23] is an extensive component library for JSF, based on the open source framework *Ajax4jsf*. It enables simplified integration of Asynchronous JavaScript and XML (AJAX) capabilities into business applications without resorting to JavaScript. RichFaces also provides a large number of ready-made components which can be easily integrated into the web interface. Furthermore a variety of predefined skins supports managing the look-and-feel of an application [23].

3.2.4 JBoss Seam

According to Yuan and Heute [24, p.3], JBoss Seam is the missing framework on top of JEE 5. It can be seen as mediator between the persistence tier, the web user interface (presentation tier) and the transitional business logic of a enterprise web application. JEE applications mostly use the core frameworks EJB3 for business logic and JSF for the web front end. JBoss Seam uses an annotation-based-approach to tie these frameworks together. Due to this approach the number of XML configurations files decreases dramatically. With the JBoss Seam framework it is possible to manage the persistence context over the entire web interaction life cycle. It is fully optimised for applications which adhere the Web 2.0 principles [24, pp.3-10].

3.3 Tools

3.3.1 AndroMDA

"*AndroMDA is an extensible generator framework that adheres to the Model Driven Architecture (MDA) paradigm*" [25]. It enables the transformation from PSM created

with UML tools into deployable components. Supported platforms are JEE, Spring and .Net. Compared to other MDA toolkits AndroMDA already offers ready-made cartridges like Hibernate, EJB, JSF, Struts and others. It offers also the possibility for building tailor-made cartridges or to modify the existing ones according to the developers needs [25].

3.3.2 AndroMDA3 Template Project

The AndroMDA3 Template Project [26, 1] developed by IGB adheres to the MDA paradigm. From a PSM UML model deployable Java code is generated using AndroMDA (sec. 3.3.1) which is built with Maven (sec. 3.3.8). Code generation is performed using two generator cartridges, JSF and EJB3, in combination with two custom cartridges, Java and Java Management Extensions (JMX).

It provides base functionality like file uploading, a mailing service, report and sharing mechanisms. A detailed description of the AndroMDA3 Template Project is given in [1]. The AndroMDA3 Template Project is used to derive the *Management Application for Mutein Data*.

Report Bean Mechanism Each entity in the UML model tagged as **Manageable** is able to provide report mechanism support. The according classes needed by the report mechanism are auto generated for each manageable entity within AndroMDA code generation. Core features are a universal list view which additionally provides customisation of the shown properties, *Edit Display Settings* and a filtering option, *Query*. This sorting mechanism allows to perform individual queries wich can be build by the user without using directly SQL. Therefore **SearchableFields**¹⁶ can be limited by using predefined **SearchTerms**¹⁷ and combined with logical operator AND. Hence, purposeful results can be retrieved. Figure 3.4 shows using the mutein list the display settings customisation (1) and query (2) panel. The filtering query is built by specifying at least one value from the *SearchableFields* (3) drop down menu in combination with an operator from the *SearchTerms* drop down list (4). The provided input field (5) within the query panel is used for specifying the filter criterion. Multiple filter criterion can be combined with AND.

Data Dictionary An integrated *Data Dictionary* (DD) allows standardised collecting and global maintaining of trivial vocabularies. Entries of the DD, termed **Lexicon** in the model, are divided into separate domains and the values within each domain are unique. The predefined values for a particular domain serve as classification aid for special properties. This list can be extended and modified according to the current needs of the user. Due to the use of key collections for different resources, consistent and standardised data entry is enabled. Hence, data analysis can be more conclusive and effective compared to arbitrary labelled data.

¹⁶SearchableFields: Properties of the entity

¹⁷like, not like, =, <>, <, >, >=, <=, between, not between, in, not in

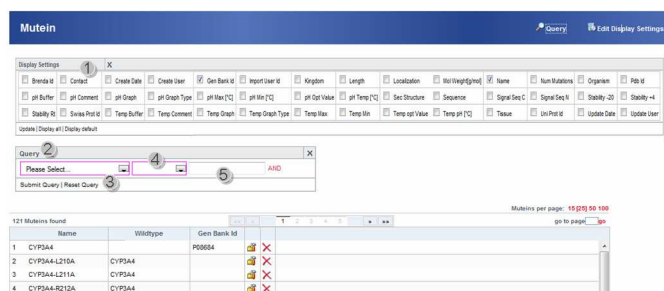


Figure 3.4: Snippet of the report bean created mutein-list view with expanded Query and Edit Display Settings panel. (1) Edit Display Settings panel, allows to customise displayed properties within the list-view. (2) Query panel, supports filtering the list, by specifying at least one SearchableField (3), a SearchTerm (4) and a criterion (5).

3.3.3 Entrez eUtils

Entrez Programming Utils (eUtils) [27] is a server-side program interface to the Entrez¹⁸ database of the NCBI. Different services are offered to query information using simple URL syntax containing the search parameters. The type of the response can be also specified with the URL parameters.

For example, as a service of the Entrez portal, PubMed can be queried by using the eUtils programming utils, in particular using ESummary engine. PubMed is a text based meta-database for biomedical literature across-the-board. The system was developed by the NCBI and is freely available via web browser as well as via defined APIs. Each record is identified by its unique PubMedID (PMID) and comprises literature meta data (author, journal, pages), abstract and a link to the full-text resource. ESummary returns a brief synopsis, termed Document Summary (DocSum), that matches the IDs specified in the request.

3.3.4 PubChem - Power User Gateway

PubChem¹⁹ [28] was founded by the National Center for Biotechnology Information²⁰ (NCBI) and is a public available database system for experimental data of small molecules and their biological activities. The system comprises three dynamically growing primary databases, *PubChem Substance*, *PubChem BioAssay* and *PubChem Compound*. The information of the three components comprises information about structural formula, identification codes, descriptions, results of bioactivity test and validated chemical information for each molecule. PubChem is also linked to other National Institute of Health²¹ (NIH) databases like PubMed. Records within the PubChem databases are identified by their BioAssy ID (AID), Substance ID (SID) or their Compound ID (CID) respectively.

CID - PubChem Compound ID PubChem Compound ID (CID) serves as unique identifier of chemical compounds within *PubChem Compound* database [29].

¹⁸<http://www.ncbi.nlm.nih.gov/sites/entrez>

¹⁹<http://pubchem.ncbi.nlm.nih.gov/>

²⁰<http://www.ncbi.nlm.nih.gov/>

²¹<http://www.nih.gov/>

Compound is one of the main three PubChem databases. It contains validated chemical descriptions which extend records contained in *Substance* database. Each record comprises chemical structure, trivial names, synonymes, molecular formula, molecular weight, acceptor count, SMILES representation, hydrogen bond donor and chemical properties as well as links to other structurally similar compounds in other NCBI databases.

The system is freely accessible via a web user interface as well as via the Power User Gateway (PUG). The Power User Gateway (PUG) [30] is an XML-based interface enabling low-level server to server access to PubChem services. The data is exchanged using XML files. All XML requests are sent to the CGI script at the URL: <http://pubchem.ncbi.nlm.nih.gov/pug/pug.cgi>

The following listing provides an overview about PUG's features:

- substance/compound download task
- chemical/substructure query task
- bio assay data query and download
- link to Entrez eUtils
- SOAP wrapper

PUG Communication As already mentioned, XML request are sent to a CGI script. In all PUG transactions the <PCT-Data> data container at the top-level is used for any PUG input or output. Users will always send an XML request with <PCT-Data> containing <PCT-InputData>, and always receive an XML response comprising of <PCT-Data> containing <PCT-OutputData> [30].

There are two scenarios after submitting a task to the PUG. Either the task is queued or it is executed immediately. Mostly the task is queued and a request ID is returned within the XML response. This ID allows to poll PUG until the task is complete. If so, the PUG returns the result in the XML response. In case of errors the response contains a simple status message describing the problem more or less meaningfully [30].

The PUG is used for querying SMILES representation of molecules by their CAS or CID. An example for a complete PUG communication is given as APPENDIX B.

3.3.5 CrossRef Meta Data Service

"CrossRef is an independent membership association founded and directed by publishers to give the research community easier access to online publications" [31].

The overall objective of the association is to provide persist and accurate links to online resources. The basic approach is the use of a Digital Object Identifier (DOI)

for managing different resources like articles, books or theses.

A DOI is a alphanumeric unique string which can be assigned to all kind of intellectual properties at any level of granularity within a digital environment. The main goal of the DOI system is to simplify managing digital objects like images, journal articles, e-books or music. This string serves as a stable and persist link to the content of the specified item. It comprises a prefix and a suffix which are separated by a slash "/". Overall number of characters which can form a DOI is limited to 128. The prefix is defined by a legalised Registration Agency (RA), which has to be a member of the International DOI Foundation (IDF). Suffix can be chosen arbitrarily by the publisher under aspects of the creation of an unique identifier. The advantage of the DOI is, that it does not change over time even if the location of the resource changes [32].

The CrossRef Meta Data Service (CMS), provides a free available experimental interface for querying meta data for a given DOI. The service is accessed via an URL request comprising of the DOI and the response format, XML. From the XML result returned, the meta data for the requested DOI can be extracted easily using an XML Parser.

3.3.6 Eclipse

Eclipse [33] is a freely available, open-source, multi-language software development environment written in Java. It contains an integrated development environment (IDE) and an extensible plug-in system. Capabilities can be extended by installing a variety of available Eclipse plug-ins like subversion control (SVN) or development toolkits for other programming languages.

3.3.7 MagicDraw UML

MagicDraw [34] is a powerful UML modelling tool. It supports the most common code engineering mechanism, database schema modelling, DDL generation and reverse engineering facilities. Due to its Java implementation it can be used on any platform supporting Java 5 onward.

3.3.8 Maven

Apache Maven [35] is a project management and built tool for Java projects. It is based on the *Project Object Model* (POM). Due to the information of the POM, maven is able to manage a project's build, documentation and reporting form a central piece of information. The primary goal of Maven is to provide developers building their projects as fast as possible.

Dependencies needed during the build process are fetched or updated from a global repository [36], and cached in the local, where also tailor-made or other additional dependencies are located. The use of a local repository decreases duration of the built-process.

3.3.9 Oracle SQL Developer

The Oracle SQL Developer [37] is a free and well documented Java tool for database developers. It provides a graphical user interface and enables the doing read/update/delete operations on database objects. Furthermore it supports testing SQL-statements and scripts, creating and debugging PL/SQL-procedures and simple database analysis.

3.4 General Nomenclatures

This section provides descriptions of different nomenclatures which are used throughout this thesis.

3.4.1 Canonicalization

Canonicalization is known in terms of generation an unique, standard, unambiguous and general valid form for the representation of a special kind of data. Reasons for canonicalization are listed below:

- check whether different representations are equal
- determine number of distinct structures
- canonicalization of filenames within file systems
- generation of canonical URLs
- create a standard sorting order

3.4.2 CAS Number

A CAS Number [38], **C**hemical **A**bstract **S**ervice Registration Number, is a numeric string (in case its not grouped by hyphens) for identifying any kind of chemical substances which was introduced by Chemical Abstract Service²². The main advantage of CAS numbering compared to empirical formula, trivial names or International Union of Pure and Applied Chemistry IUPAC names is, that a CAS number servers as unique identifier for known chemical compounds, elements, compounds, biological sequences, polymers, mixtures and alloys. Different isomers of a compound have their own CAS Number.

Integrity of CAS numbers can be checked easily by calculating the check sum digit of the ID. CAS numbers includes up to ten digits which are grouped in three divisions separated by a minus symbol "-", see fig. 3.5. The first section, from left to right, has up to seven digits, the second - middle part comprises exact two digits and the last section contains one single digit, representing the check sum. For CAS number validation all digits, except check sum digit, are taken into account [39]. CAS-Number structure is described in fig. 3.5. The formula for the integrity calculation is given

$$\begin{array}{ccccccc}
 N & \dots & N & N & - & N & N & - & R \\
 i & & 4 & 3 & & 2 & 1 & & \\
 \\
 N & \dots & \text{fundamental sequential number} \\
 R & \dots & \text{check sum digit}
 \end{array}$$

Figure 3.5: The CAS-Number comprises three main parts (starting from the left): segment of up to seven digits separated by "-" from the second part, two digits, again separated by "-" from the check sum digit (R). Maximal number of digits (N) is ten.

as equation 3.1.

$$\frac{iN + \dots + 4N + 3N + 2N + 1N}{10} = Q + \frac{R}{10} \quad (3.1)$$

N ... fundamental sequential number
Q ... discarded digit
R ... check sum digit

Example: Integrity check for the molecule phenylacetone (103-79-7), see equation 3.2.

$$\begin{aligned}
 \frac{5 * 1 + 4 * 0 + 3 * 3 + 2 * 7 + 1 * 9}{10} &= 3 + \frac{7}{10} \\
 \frac{37}{10} &= 3 + \frac{7}{10}
 \end{aligned} \quad (3.2)$$

3.4.3 EC Number

The Enzyme Commission Numbers, EC-Numbers [40] are a numeric classification system for enzymes, whereas the number is used to categorises enzyme-catalysed reactions and not the enzyme itself. Each number starts which EC followed by four numeric values separated by a dot. The first number servers as base classifier. The more right the value is located the more detailed is its meaning. Catalytic reactions are divided into six major classes, shown in table 3.1.

3.4.4 Simplified Molecular Input Line Entry Specification

Simplified Molecular Input Line Entry Specification (SMILES) [41] notation is a one dimensional string representation of a two dimensional chemical structure. Since it was introduced in 1988 by Weininger [41] it has found widespread acceptance. The advantage of SMILES compared to the former used Wiswesser Line Notation (WLN) [42], is that it is much easier to read and use because of its few generation rules.

Generation Rules

The following sections introduce SMILES generation rules according to Weiniger [41]. This specification is also used by the SMILES generation function of the CDK see sec.

²²<http://www.cas.org/>

EC Class	Name
EC 1	Oxidoreductases
EC 2	Transferases
EC 3	Hydrolases
EC 4	Lyases
EC 5	Isomerases
EC 6	Ligases

Table 3.1: Tabular listing of the six main enzymatic groups used for describing enzyme-catalysed reactions.

3.5.3. In SMILES representation default related hydrogens are not taken into account.

Atoms In SMILES, they are represented by their atomic symbol. Aliphatic atoms are represented by upper case symbols, aromatic atoms by lower case symbols [41]. See table 3.2 for some examples.

Molecule	Chemical formula	SMILES
methane	CH ₄	C
phosphine	PH ₃	P
ammonia	NH ₃	N
hydrochloric acid	HCl	Cl

Table 3.2: Examples for atom SMILES representation.

Atoms which are not in the organic subset or with abnormal valences need to be described in brackets. *Example:* elemental Gold [Au]. Formal charge can be shown within brackets by + and - respectively. Charges and number of hydrogens are assumed to be zero if not otherwise specified [41]. Table 3.3 provides some examples.

Bonds The following listing represents the four different bond types which are defined within the SMILES specification rules. Use case examples are given in table 3.4. Single and aromatic bonds are usually omitted.

- - single bond
- = double bonds

Molecule	SMILES
proton	[H ⁺]
hydroxyl anion	[OH ⁻]
iron(II) cation	[Fe ²⁺]
iron(II) cation	[Fe ⁺⁺]

Table 3.3: Examples for atom SMILES representation with formal charge.

- # triple bonds
- : aromatic bonds

Molecule	Chemical Formula	SMILES
ethane	(CH ₃ CH ₃)	CC
ethylene	(CH ₂ =CH ₂)	C=C
hydrogen cyanide	(HCN)	C#N
molecular hydrogen	(H ₂)	[H][H]
dimethyl ether	(CH ₃ OCH ₃)	COC

Table 3.4: Examples for bond generation rules of SMILES representation.

SMILES representations are not unique by default. There are multiple ways for representing a single structure. For example the structure for 6-hydroxy-1,4-hexadiene, see table 3.5, can be represented by three different SMILES strings. All of these are valid line representations of the structure. See section 3.5 for generating unambiguous SMILES for a given structure.

Branches Generating branches is very simple. It is done by enclosures in parenthesis. Branching examples are given in table 3.6

Cycles Presentation of cyclic structures is done by breaking up cycles and presenting it as non cyclic graph. This graph is used for creating the SMILES by marking

Chemical formula	SMILES
CH ₂ =CH-CH=CH-CH ₂ -OH	C=CCC=CCO
	C=C-C-C=C-C-O
	OCC=CCC=C

Table 3.5: Multiple SMILES representing the structure of 6-hydroxy-1,4-hexadiene.

Molecule	Structural formula	SMILES
triethylamine	$ \begin{array}{c} \text{CH}_3 \\ \\ \text{CH}_2 \\ \\ \text{H}_3\text{C}-\text{CH}_2-\text{N}-\text{CH}_2-\text{CH}_3 \end{array} $	<chem>CCN(CC)CC</chem>
isobutyric acid	$ \begin{array}{c} \text{CH}_3 \quad \text{O} \\ \quad \\ \text{H}_3\text{C}-\text{CH}-\text{C}-\text{OH} \end{array} $	<chem>CC(C)C(=O)O</chem>

Table 3.6: Examples of branch representation in SMILES.

a loop at the beginning and the end with the same integer number. This enables to represent multiple cyclic structures withing the same molecule [41].

Disconnected Structures Also a rule for the representation of the disconnected structures is provided by the Weininger [41] definition. They are treated as individual structures separated by a dot "." [41].

Aromaticity Organic compounds which contain an *aromatic ring* within their structure are called aromatic compounds. As already mentioned within the generation rules for atoms, aromaticity is denoted by lower case symbols. In case of aromatic structures, atoms that belong to a aromatic ring, are written in lower case letters. Table 3.7 illustrates representation of the aromatic compound *Aspirin*. The according aromatic SMILES is CC(=O)Oc1ccccc1C(=O)O [41].

There also exists other, or extended generation rules for SMILES which resulted from different needs. More details can be found at the Daylight SMILES theory tutorial [43].

Unique SMILES

As already mentioned in section 3.4.4 the unique structure of single molecule can be represented by multiple SMILES. Because of the need to distinguish between different SMILES without generating and comparing all possibilities, rules for the generation of a **unambiguous** SMILES, the so called canonical or unique SMILES, were described

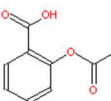
Molecule	Molecular Formula	Structural Formula	SMILES
Aspirin	$\text{C}_9\text{H}_8\text{O}_4$		<chem>CC(=O)Oc1ccccc1C(=O)O</chem>

Table 3.7: Aspirin as example for an aromatic compound. The example give structural and molecular formula as well as SMILES presentation of Aspirin. Atoms which are part fo the aromatic ring are written as lower case letter, highlighted grey within the SMILES

by Weininger [44].

The base idea of creating a unique SMILES is the **canonical order** of the atoms. The method introduced by Weininger [44] is accomplished in a two-stage algorithm termed **CANGEN**. It comprises the combination of **CANON** and **GENES** algorithms.

CANGEN First Stage The first stage of the CANGEN algorithm involves canonicalization of the molecule structure. This is done by the **CANON** algorithm which label a molecular structure with canonical labels. **CANON** algorithm follows closely the widely known canonicalization approach introduced by Morgan [45]. Canonicalization process is illustrated in fig. 3.7. The iterative procedure starts with assigning the number of connected atoms to each atom node. Maximum connectivity value is represented by n , $n=3$ after finishing first iteration. During the following subsequent iteration steps connectivity count for each atom node is recalculated until the maximum connectivity value is reached. This means the iteration stops if $n = n-1$. In the given example this state is reached after five iterations [44].

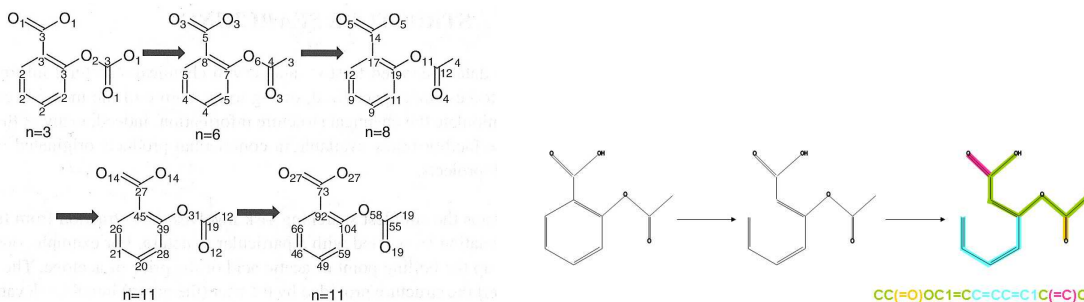


Figure 3.6: Canonical labelling of a molecule structure for generating unique SMILES [46]. Beginning with assigning the number of connected atoms to each atom node. In the following subsequent iteration steps this number is recalculated. The iteration procedure stops if the maximum connectivity value is reached for each atom.

Figure 3.7: Unique SMILES generation example for aspirin. To result a non cyclic graph all cyclic structures have to be broken up first. Beginning at the lowest connectivity index of this graph the SMILES is written in branches.

GENES Second Stage **GENES** algorithm is used to create a molecular graph starting at the lowest ranked atom. According to the SMILES notation and the created molecular tree the unique SMILES is generated.

Figure 3.6 illustrates unique smiles generation of aspirin (canonical labeling see fig. 3.6). Beginning with breaking cyclic structures of the molecule the non cyclic graph is created. According to the canonical numbers SMILES generation is started at the lowest number [44].

3.5 Libraries

3.5.1 Apache POI

Apache POI - the Java API for Microsoft Documents is a compound of Java libraries for opening, reading, modifying and writing on Microsoft Office Formats [47].

Horrible Spread Sheet Format (HSSF)

The HSSF library enables reading and writing from and to MS Excel '97(-2007) file formats [47].

3.5.2 Genome Usermanagement

The User Management developed by IGB²³ [48, 49], is a web-based user management and Authentication and Authorisation System (AAS) which is able to manage all resources and uses assigned to an application. Settings can be created and modified via the web interface of the user management server by authorised users. The main task of the user management is the registration of new and the management of existing users and user groups, applications and their resources and access levels. An Example for the use of the `aas:permission` tag within a JSF page is given in APPENDIX C.

3.5.3 Chemical Development Kit

The Chemistry Development Kit (CDK) [50] is a collection of Java libraries which support various computational tasks in structural chemo-, molecular- and bioinformatics. As an open source project it is freely available from SourceForge²⁴. The CDK comprises simple I/O routines, mechanism for parsing and generating SMILES, rendering two and three dimensional structures up to complex isomorphism testing algorithms [51].

The following sections describe the main CDK classes which are used for implementing substructure search feature in more detail.

Fingerprinter

A fingerprint is the abstract one-dimensional bit representation of a two-dimensional structure. Because there is no special meaning to a set bit structure can not be restored from a given fingerprint. Set bits represent the occurrence of a particular structural feature [52].

Whereas a bit not set represents the absence of a structural feature, a bit set indicates its occurrence not for sure.

Fingerprints are often used for pre-screening in substructure search because "*A pattern is a substructure of a molecule, if every bit that is set in the pattern's fingerprint also is set in the molecule's fingerprint*" [53].

CDK Fingerprint Algorithm The implementation of the CDK fingerprint algorithms follows closely the Daylight [53] approach. The algorithm first performs a depth first search looking for all atom sequences up to six atoms. For all this generated sequence snips, a hash code is generated, using the build-in Java hash function. These hash codes are used for initialising a pseudo-random number generator (RNG),

²³Institute for Genomic and Bioinformatics, Technical University Graz

²⁴<http://sourceforge.net/projects/cdk/>

which returns the first random number between 0 and 1023. The RNG result is finally taken to set the respective bit in the fingerprint array. Using hash code always results in information loss. So different molecules can result in the same fingerprint representation [51]. The correctness of this method in distinguishing different structures has been shown by Brown [54].

Molecule

`org.openscience.cdk.Molecule` maps the concept of molecule structure. A `org.openscience.cdk.Molecule` object comprises atoms connected by bonds. In combination with the `SmilesParser` a molecule is created out of the structural information represented by the SMILES [52].

SubgraphIsomorphismTester

The `SubgraphIsomorphismTester` contained in the `org.openscience.cdk.isomorphism` package provides three structural tools:

- maximum common substructure (MSC) searching
- substructure testing
- mapping of two isomorphic structures

The code example in listing 3.1, gives a simple example of a substructure search using the `UniversalIsomorphismTester`. Substructure searching of the CDK is based on `RGraph` (Resolution Graph) class. The algorithm implemented by `RGraph` is derived from the maximum common substructure algorithm described in [55].

```
1 SmilesParser sp = new SmilesParser(DefaultChemObjectBuilder.getInstance());
2 IAtomContainer atomContainer = sp.parseSmiles("CC(=O)OC(=O)C");
3 IAtomContainer SMILESquery = sp.parseSmiles("CC");
4 IQueryAtomContainer query = IQueryAtomContainerCreator
5     .createBasicQueryContainer(SMILESquery);
6 boolean isSubstructure = UniversalIsomorphismTester.isSubgraph(atomContainer,
    query);
```

Listing 3.1: Code snippet UniversalIsomorphismTester

SmilesParser

`org.openscience.cdk.SmilesParser` decomposes a SMILES string according to the SMILES generation rules in [41] and returns a `Molecule` or any other `AtomContainer` [52].

3.5.4 JDOM

JDOM is light weighted and straightforward Java library for manipulating XML documents. It provides simplified solutions for accessing, manipulating and outputting XML data from Java code. [56].

3.5.5 overLIB

overLIB is a JavaScript library which simplifies the integration of small popup information boxes into websites. Mostly it is used to provide additional information or instructions for a particular component [57].

3.5.6 JMEMolecularEditor[©]

The JMEMolecularEditor [58] is a compact Java applet written by Peter Ertl, Novartis Institutes for BioMedical Research, Basel, Switzerland. It is freely available for non commercial use under some particular agreements. The license agreement is given as APPENDIX A. The applet allows to draw, edit and display molecule and reaction information within a web page. Neither calculations nor database search operations can be done by the JME. In the particular case of the MuteinDB it is used for drawing the target pattern in case of substructure search and for displaying this pattern on the result page.

Integration of the applet into the application is very simple. The integration code snippet is provided in listing 3.2. Usage in a web browser requires installing and enabling Java in the browser options [58].

The functions provided by JME can be accessed by JavaScript (JS) on the web page. All functions and further information about JME can be found at the Molinspiration²⁵ webpage.

```
1 <h:form id="jme">
2   ...
3   <applet code="JME.class" name="JME" archive="JME.jar"
4     codebase="/muteindb-web-0.2/applet" width="380" height="355">
5
6     <param name="options" value="query, xbutton, rbutton" />
7
8     To use JME enable Java in your browser options!
9   </applet>
10   ...
11 </h:form>
```

Listing 3.2: HTML snippet for integrating the JME editor into the web front end.

JME Features

Figure 3.8 shows the plain JME Editor window. The top menu comprises available features. These properties can be modified by the editor option attribute. The following listing gives an overview about all available editor features.

- **SMILIE** show smiles string for editor content
- **D-R** deletion of functional groups
- **CLR** clear whole editor window
- **UDO** reverse last change (undo)
- **X Button**

²⁵<http://www.molinspiration.com/jme/>

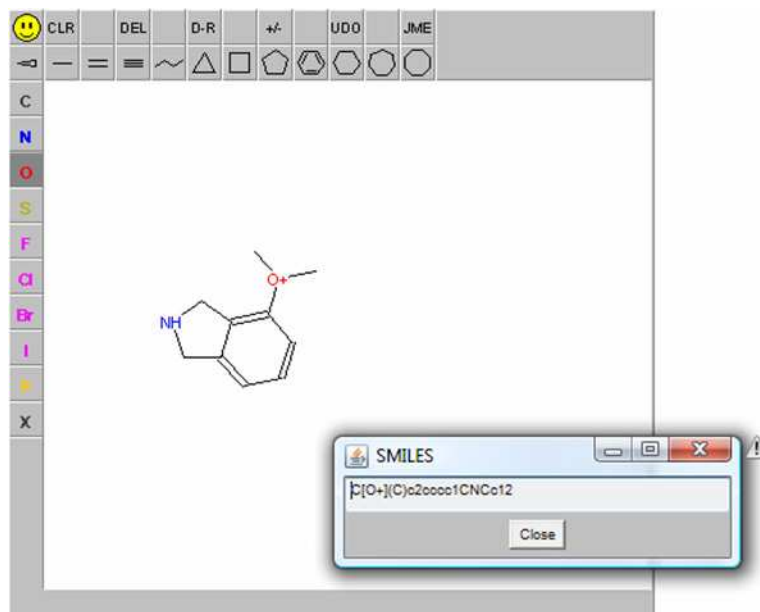


Figure 3.8: Illustration of the JME Editor. Atom and structure shaps are situated left and above of the drawing window. The main top menu contains enabled features of the editor. SMILES representation is displayed by clicking on the smily, in the left upper corner, in a new window.

- +/- modification of atomic charges
- NEW entering multipart structures
- 123 mark atoms
- QRY allows to create structures containing wildcards
- END
- JME show "About" information

At the left hand side and at under the top main menu the shape and molecule tools are located. **Note:** Hydrogens are implied in the structure representation.

Public Functions

public String JME.smiles() returns the unique SMILES of the molecule/pattern drawn in the editor window. SMILES generation is done as specified by the Daylight Chemical Information Systems, Inc.²⁶.

public String JME.jmeFile() returns a string representation of the current JME content which can be used for initialising the editor in display mode.

²⁶<http://www.daylight.com/>

`public void JME.options(String)` allows to modify the applet parameters. The function takes a comma separated list of keywords which are used to change the applet parameters dynamically from JavaScript. Details about keywords of interest are provided below. A full list of all keywords can be found in the JME Basic Instructions²⁷.

- **nostereo** take no account of stereochemistry
- **multipart** enable multipart structure drawing
- **number** mark atoms
- **depict** applet in display mode

²⁷http://www.molinspiration.com/jme/doc/jme_help.html

Chapter 4

Results

In the course of this thesis, a web application for the management of mutein data has been developed, under the aspects of model driven software development. The application enables mechanism for data collection, storage, searching and presentation. It is currently used by the research group of the Institute of Molecular Biotechnology (IMBT) at the Technical University Graz (TUG). At the present time the application is used for internal research and analyses only. In the future it is planned to provide full access for members of the EU OXYGREEN²⁸ consortium and read access to selected other.

This chapter describes the web application, which was implemented during the work on the diploma thesis. According to the tree-tier-architecture the chapter is divided into three main sections. The focus of the first section is on the development of the database schema. The second describes the business logic of the application. The third and last section illustrates the web front end, the user interface of the application.

The implementation is based on user requirements collected by Michael Gugge-mos in the course of his diploma thesis [15], which have been modified extensively in discussion with users of the research group of IMBT at TUG.

4.1 Database Structure

The database structure of MuteinDB 2.0 is based on the schema of the prototype version. In several meetings with members of the research group of the Institute of Molecular Biotechnology relations and attributes of the schema were redesigned. As not all needed relations were taken into account in the former schema, and related to the extensions of features, considerable changes were required. An overview of the tables and relations of the prototype database schema can be found as APPENDIX D.1.

²⁸<http://www.oxygreen.org/>

4.2 Development of UML Diagram

According to the modified database schema the UML entity diagram of the application was designed. The UML model of MuteinDB 2.0 is given as APPENDIX D.2. The model comprises following ten relational entities describing proteins, muteins and their catalytic activity.

Mutein contains wildtype protein as well as mutein entries. The table contains beside basic information like create date, user name, contact data also enzyme property values like organism, tissue, pH or temperature values.

Mutation table references to **Mutein** and contains at least one mutation for each mutein entry. A mutation comprises the original, the modified AA(s) and the correlated position in the sequence and the mutation type. Only mutations at the AA level are taken into account. The form of the mutation results from the set values of a single entity.

PosTranslational - Post Translational Modification (PTM) allows to register chemical modification of proteins after finished protein synthesis. The table represents a n to n relation between a mutein entry with the correlated lexicon foreign key (FK) of the PTM.

Inhibition contains data about related inhibitor molecules to a protein or mutein entry, like the used buffer or solvent, temperature and others. The table also references to the literature source entry from which out the data was collected.

Catalysis table references in each case to one substrate and product molecule in **Molecule** as well as to related activity entries. Also reaction type and enzymatic class membership of the enzymatic reaction is stored.

Molecule contains information about substrates, products and inhibitors. The molecule number (CID and/or CAS), name and structure as SMILES and fingerprint is stored for each entry.

Activity contains information about parameters under which the specific activity of a catalytic reaction was measured. For muteins also the reference activity of the wildtype protein is registered. The table references to the according literature entry from which out the information was collected.

CoFactor contains one or more co-factors references related to an activity. The table represents a $n:n$ relation between an activity entry, with the correlated lexicon FK of the co-factor.

CoProtein contains one or more co-proteins references related to an activity. The table represents a $n:n$ relation between an activity entry, with the correlated lexicon FK of the co-protein.

Source is used to provide information about the source - a publication - of the collected enzyme data. It contains meta data like author, title, journal and others for the literature reference of an enzyme.

All ten entities (tables) are represented by an Entity Bean, a Session Bean and a global Exception Class.

4.3 Business Logic

According to the use of JEE, business logic is encapsulated in EJB3. Due to code generation with AndromDA, Create **R**ead **U**pdate **D**ele~~t~~e - CRUD - business functions are generated automatically for new **manageable** entities. Business functions which were needed by new features were modelled as SessionBean (tagged by **Service** in the UML model).

The following *SessionBeans* have been defined within the management application for muterin data:

DataImportService provides services used by the automatic data import from MS Excel files.

ExternalService provides services for retrieving data from external databases like GenBank, PubMed, CrossRef or PubChem.

SearchService provides services for the different search mechanisms.

StructureSearchService contains services related to substructure searching mechanism.

Furthermore for all this additional Session Beans the related Exception class has been modelled.

4.3.1 Data Import Service

The data import enables registering sets of muterin data automatically by uploading the collected data using a Microsoft (MS) Excel file. A simple upload interface is used for uploading the file and starting the reading process at the same time. During the data import process the file is cached to a temporary directory and removed after the process has finished.

In collaboration with members of the research group of the Institute for Molecular Biotechnology the existing MS Excel file was revised and unified so it can be used as template for the automatic data import. Only MS Excel files which adhere the template style are accepted by the upload interface. The template import file comprises nine different units (first row, see fig. 4.1), which describe the basic data as well as the experimental set up under which the special activity of the muterin or

wildtype protein was measured. The detailed MS Excel template sheet is provided in APPENDIX F.

Basic Data					SignalSequences			pH			
UserName	Date	MuteinName	WildtypeName	Mutations	N-Terminal	C-Terminal	pHMin	pHMax	OptimumpH	Temp [°C]	Buff
Weinhandl	20090714		CYP3A4								
Weinhandl	20090714		CYP3A4								
Weinhandl	20090716	CYP3A4-L210A	CYP3A4	L210A							
Weinhandl	20090716	CYP3A4-L210A	CYP3A4	L210A							
Weinhandl	20090716	CYP3A4-R212A	CYP3A4	R212A							
Weinhandl	20090716	CYP3A4-F213A	CYP3A4	F213A							
Weinhandl	20090717	CYP3A4-F304A-His	CYP3A4	F304A, CTAGHHHH							
Weinhandl	20090717	CYP3A4-L211F/D214E-His	CYP3A4	L211F, D214E, CTAGHHHH							
Weinhandl	20090717	CYP3A4-His	CYP3A4	CTAGHHHH							
Weinhandl	20090818	CYP3A4-His	CYP3A4	CTAGHHHH							
Weinhandl	20090819	CYP3A4-A305F-His	CYP3A4	A305F, CTAGHHHH							
Weinhandl	20090819	CYP3A4-A305S-His	CYP3A4	A305S, CTAGHHHH							
Weinhandl	20100212	YP3A4-L216W/F228I/T433S-Hi	CYP3A4	N, F228I, T433S, CTAGHHHH							

Figure 4.1: Segment of MS Excel data import template. First row (grey) contains group names. The second row (green) comprises the properties of each group.

Basic Data contains information about when and by whom the data was collected as well as fundamental features like organism, tissue, sequence, in case of mutein the wildtype protein and relations to entries in other databases.

Signal Sequences contains data about short AA sequence snippets at the beginning or end of the sequence.

pH Range comprises minimal, maximal and optimal, as well as pH value related information for determining the preferred pH value environment of the mutein.

Temperature Range comprises minimal, maximal and optimal, as well as temperature related information for determining the preferred temperature environment of the mutein.

Storage Stability describes storage stability of the enzyme at room temperature, minus 20°C and plus 4°C.

Inhibitor contains data about the involved inhibitor molecule, the degree of inhibition and the set-up.

Reaction comprises the main parameters which describe a reaction, substrate and product. Additionally data about the enzymatic class (EC-number) and the type of the reaction are registered.

Activity characterises the outcome of the substrate specific reaction as well as the literature source of the measured values.

Reference Activity in case of mutain entries additionally to each activity of the mutain the activity of the wildtype protein is registered.

The second row, highlighted green in fig. 4.1, of the MS Excel file contains the different parameters which belong to the same group. A Segment of the import template is given as fig. 4.1.

The combination of group and column name is used to create a unique label for each column. These unique column names are necessary for form validation and finding the corresponding column number in the import file. Validation within the data import comprises three main phases, *type*, *form* and *value* validation, see fig. 4.2. A positive type check initialise the form validation of the file. First, the number of columns is counted. If the number of columns exceeds the expected count the unique labels are compared to the template. If all column names are present in the uploaded file the data parsing is started. This implies that group and column name of the used import file are equal to the required defined template. Otherwise the form validation fails.

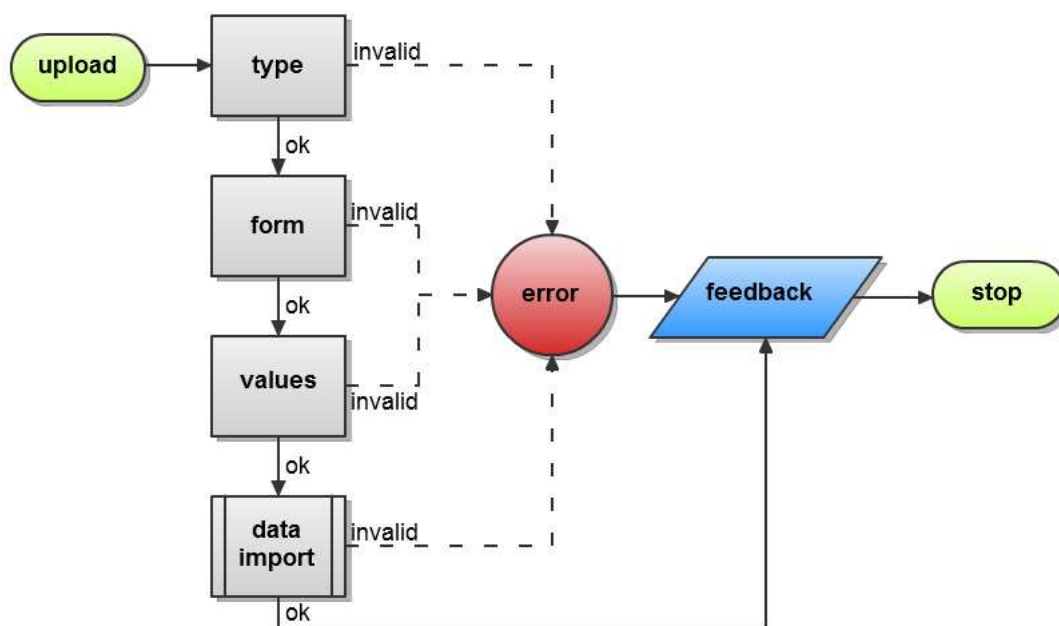


Figure 4.2: Data is validated in three steps during the import procedure. First file type is checked, followed by the validation of template form. Finally every single cell value is audited before the data is written to database.

The data is read row by row and column by column and starts at line three. During the import, a *value tree* is build from the Excel data. This data structure is an hierarchical representation of the enzyme data in the Excel sheet according to its relations. Depending on the column label and row type, mutain or wildtype, cell val-

ues are checked in various ways. There are three main cell value validation methods. In some cases all of them are applied to a single cell object. Figure 4.3 outlines the main three cell value validation mechanisms. Every cell is checked whether a value is mandatory or not and if the value is of correct type. In case of missing values or incorrect types an error message is passed to the `ErrorMessageHandler`. Furthermore for some values integrity checks are done. For example, a cell containing mutation data, the nomenclature and the correctness of the position in the original sequence is checked. If errors occurred during the value tree building phase the error report is displayed via the feedback web interface, see sec. 4.27, without creating any database entries.

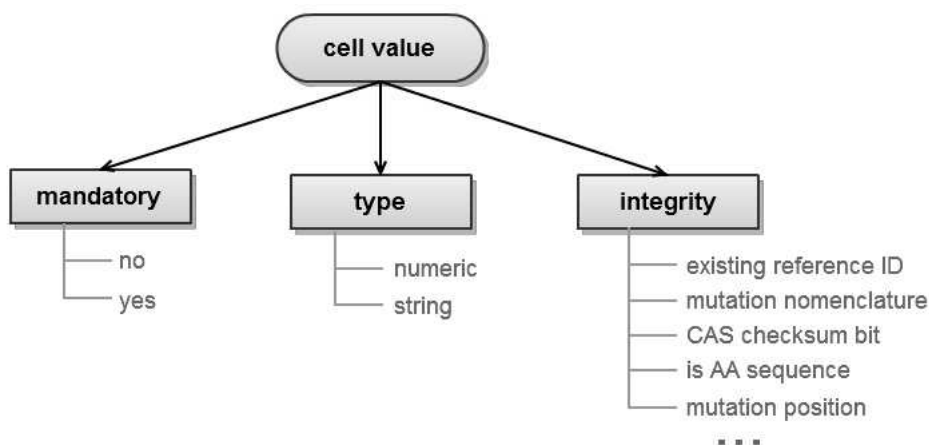


Figure 4.3: Illustration of the three main cell value validation methods, applied during the data import. Cells can be mandatory and/or restricted to either numeric or string values. Also mechanisms for integrity checks are provided.

In case of successful parsing a complete value tree is created. Based on the bottom-up principle (beginning from nodes moving to the root of the value tree), the data is written to database. In this way references are created before they are needed. The whole persisting procedure is done within the same transaction. This guarantees that only complete value trees are written to database. In case of errors within the persisting process the transaction is marked for rollback, which resets the database to the state before the start of transaction.

Guidelines for the data collection using the defined MS Excel template was created. These guidelines comprise mandatory fields, nomenclatures for mutation and mutein names, unit standards for activity, inhibition, temperature and time values as well as rules which identifiers to use for molecules and literature references. These standards are the fundamental element of the integrity check methods of the cell value verification mechanisms. The complete guidelines are available as APPENDIX G.

Special Data Import Mechanism

During the data import some of the values are used to query information or meta data from external databases. This meta data is set to the correlated object properties

and written into database within the import. Additionally heavy use is made of the integrated Data Dictionary to guarantee uniform and consistent attribute values. Values which are assigned as data dictionary entries are cross checked if they already exists or if they had to be created within the import procedure. Furthermore the check-sum of every CAS-number is calculated and validated according to the rules given in section 3.4.2. The following sections discuss these mechanism in more detail.

Data retrieval with GenBankId Sequence information for a wildtype protein or a muterin is specified by the respective wildtype *GenBankId*. The request URL for the sequence data is formed according to the eUtils URL standards²⁹. Listing 4.1 shows the request for GenBankId P08684 for enzyme CYP3A4.

```
1 http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=protein&rettype=fasta&retmode=xml&id=P08684
```

Listing 4.1: eUtils request URL for sequence data of GenBankId P08684 (CYP3A4)

Additionally to the target query ID, the URL comprises the CGI script to which the request should be posted, *efetch.fcgi*, the target database *db=protein*, the response type *rettype=fasta* and *-mode retmode=xml*. The result of the request is the XML file illustrated as APPENDIX E.1. Using the JDOM XML parser, sequence, sequence length, sequence type, and organism information is extracted and set of the corresponding enzyme entry.

Structure information import from PubChem Substructure search requires canonical SMILES and fingerprint representation of the molecule. Based on molecule identification, CAS or CID, the one dimensional structure representation, SMILES, can be retrieved from PubChem using PUG. Due to the PUG restriction to CIDs, all CAS-numbers have to be transformed into the respective CID first. eSearch, a service provided by Entrez eUtils is used for CAS to CID transformation. The conversion is done by posting the according URL request, see listing 4.2, to the eSearch CGI script. In addition to the CAS number the URL comprises the target database *db=pccompound* and the number of maximal results *retmax=100*.

```
1 http://www.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=pccompound&retmax=100&term=103-79-7
```

Listing 4.2: eUtils URL for transforming CAS (103-79-7) into CID

The XML result, given as APPENDIX E.2, is parsed using JDOM for extracting the information of interest. In many cases the response contains more than one CID for a CAS number. This is caused by different physical configurations of the same molecule. As atom arrangement is not taken into account in canonical SMILES generation, the first result can be assigned to the respective CAS. If the CAS cannot be mapped to a CID, a warning is written to the `ErrorMessageHandler`. The molecule is discarded for further processing but it is written into database.

²⁹http://eutils.ncbi.nlm.nih.gov/corehtml/query/static/efetchseq_help.html

All available CIDs are submitted in groups of sixty items per XML request, given as APPENDIX B, listing B.1, to the PUG for retrieving SMILES representation. A detailed description of PUG conversation is given in section 3.3.4. The XML response is parsed using JDOM for extracting the relevant information.

Literature Meta Data Retrieval Source for collected enzyme data has to be specified either by PubMedID or by DOI. For both identifiers meta data is extracted from external databases using different services. In both cases the identifier is part of an URL request which is posted to a script. The result is returned in form of an XML response. Relevant meta data is extracted according to the XML structure of the document using JDOM.

Literature Meta Data Retrieval By PubMedID Listing 4.3 illustrates the URL for querying literature meta data from PubMed using PMID 10191269. It specifies the search engine *esummary.fcgi*, the database *db=PubMed* which has to be queried, the return mode of the result *retmode=xml* and the target ID *id=10191269*. The complete XML response is given as APPENDIX E.3, listing E.3.

```
1 http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esummary.fcgi?db=PubMed&retmode=xml&id=10191269
```

Listing 4.3: eUtils URL for querying meta data from PubChem

Literature Meta Data Retrieval By DOI Using the CrossRef Meta Data Service, described in section 3.3.5, available meta data for a publication can be retrieved with the related DOI. The request URL is shown in listing 4.4 for DOI 10.1002/adsc.200505069. It is formed by a combination of the target DOI and the desired response return type (XML). XML response of the CMS is given in APPENDIX E.4.

```
1 http://api.labs.crossref.org/10.1002/adsc.200505069.xml
```

Listing 4.4: CrossRef Meta Data Service URL

4.3.2 Structure Search Service

The structure search mechanism enables searching formolecules (substrates, products and inhibitors), which contain a particular structure pattern. The result of the substructure search, see sec. 4.4.3, shows a tabular listing of molecules in which the pattern was found.

Based on the approach described by Kai [59] substructure search was integrated. The main three mechanism needed for performing substructure searches are:

1. SMILES
2. Fingerprints

3. Graph based structure comparison.

Drawing a structural pattern is enabled by the JMEMolecularEditor Java applet, described in section 3.5.6. A form, **StructureSearchForm** and a controller class **Structure-SearchController** were implemented for managing the view. The business functions for substructure searching are provided by the **StructureSearchService**. Figure 4.4 illustrates the work flow during the substructure search procedure. The drawn structure (pattern) is converted to SMILES representation according to the Daylight generation rules. The one dimensional line notation is then used for the calculation of the bit representation of the structure. This so called fingerprint, is calculated using the CDK **Fingerprinter** class. The fingerprint is used to retrieve a preselection of possible molecule candidates in which the searched pattern is present with a certain likelihood. Set bits of the pattern fingerprint are compared to all molecule fingerprints of the database. Only molecules which contain the same bits set compared to the fingerprint are added to the preselection set. More than 90 % of the molecules are discarded during this step. The remaining molecules are submitted to the time consuming isomorphism test. The application uses the CDK **UniversalIsomorphismTester** which adheres the maximal common subgraph detecting method described by Tonnelier [55].

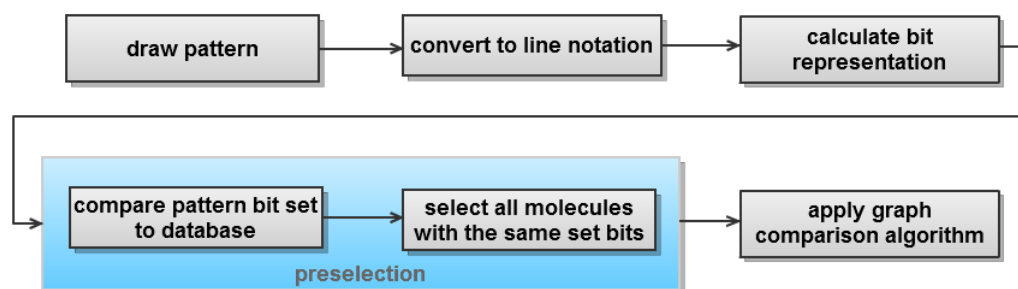


Figure 4.4: First the target pattern has to be transferred into line notation. From this representation a bit string is calculated which represents the pattern. With this pattern a preselection of the molecules is chosen. Finally an isomorphism algorithm is applied to the preselected molecule set.

4.3.3 Search Service

The purpose of all provided search mechanisms is the location of particular enzyme data. In case of *Show All* and *Search by Wildtype* the result presentation comprises enzyme related data only, see sec. 4.4.3. Search result presentation of *Search Mutein*, *Search by Reaction*, and *Search by Inhibitor* is formed by enzyme as well as by reaction, and activity data, see sec. 4.4.3. The focus of the shown result list is on the mutein entry with the highest specific activity value. Additional activities for the found enzyme is provided on demand.

For all search interfaces, (excluding ShowAll, which is generated automatically), one common form, *SearchMuteinForm*, and controller class, **SearchMuteinController**,

has been implemented. Furthermore, to benefit from `ReportBean` mechanisms the required `SearchMuteinReportService`, `SearchMuteinReportServiceBase` as well as the needed `SearchMuteinDataProvider` and `SearchMuteinDataModel` have been implemented.

The individual search query is build from the selected parameters on the respective web interface from the `SearchQueryBuilder`. This query is submitted to the `SearchMuteinReportService` to perform the database search. By default the result would consist of mutein entities. According to the multiplicity dependencies to catalysis and activity table for each match in each table the mutein entity is returned. Thus, the resulting mutein entries have to be grouped. Nevertheless, the provided queries allow to specify catalysis and activity properties as search parameters. Therefore the related catalysis and activity information for the mutein is needed. To avoid redundant database queries the result list content was changed from entities to IDs (PKs from mutein, catalysis and activity). Thus, a result list contains the number of found *Mutein* entries primary keys (PK) with related `Catalysis` PKs and `Activity` PKs, according to the selected page size. This result ID list also contains duplicated mutein PK but now they are much more easier to remove and related catalyses and activity PKs can be grouped according to the mutein entry. From this result list a hash table which maps the relations between `Mutein - Catalysis - Activity` is built. This table enables efficient on demand loading of additional activities of an enzyme into the existing data model.

4.4 Web Application

The web front end of MuteinDB 2.0 is based on the look-and-feel designed in the course of developing MARS³⁰ web application at IGB [49]. That implies the use of the defined application frameset (see fig. 4.5), style sheets, menu navigation and the user management. The main benefit related to the AndroMDA3 template project are the provided **Report Beans**, which automatically generate list, search-, and editable web views for manageable entities. This means the JSF page as well as the business functionality is generated automatically.

4.4.1 Application Look-And-Feel

Application design is based on the framework developed under MARS. It comprises four main frames, red boxes in fig. 4.5, whereby the centre-frame is formed by several subframes, green boxes in fig. 4.5. On the top of the centre-frame (4), the general top-wide frame (1), containing the application and association name, and the public available top menu-frame (2) are located. Beyond the centre-frame a footer (3) describing copyright holders and imprint is situated. The centre-frame itself comprises the content frame (8) which is surrounded by navigation menu (5), association reference (6), log in/log out facility and user information (6) and menu administration features which enables to customise the location of the main application menu. In the

³⁰Microarray Analysis and Retrieval System

following, web interface descriptions will focus on the content-frame of the application and hence other frames are not shown in the illustrations.



Figure 4.5: Red boxes: (1) Top-wide frame, contains company and application name. (2) Top-menu frame, contains web pages that have public access. (3) Bottom frame, contains information about copyright holders and the imprint. (4) Centre (main) frame. It comprises five subframes, green boxes. Green boxes: (5) Menu-frame comprise application navigation. (6) Notice frame, contains company name and logo. (6) User frame, contains login option and name of logged in user. (7) Menu-administration frame comprises options for customising location of the main-menu. (8) Content-frame, contains the current view.

4.4.2 Search Interface

The following sections describe the six search mechanisms provided by MuteinDB. The web interfaces for the search mechanisms were created manually according to style rules, defined by the template project, to adhere to the IGB look-and-feel.

Basic Selection Schema

All search masks adhere to a basic value selection schema. A search parameter comprises four elements, name (1), selecting-aid (2), value list box (3) and exclude check box (4), see fig. 4.6. All values which are available for the parameter termed in (1) are listed in the box (3) alphabetically. The value can be selected by browsing through the box or by using the input field (2). By typing into the input field the value most similar to a box value is selected or (*none*) in case of no match. The selection can be inverted by tagging the *exclude* check box (4). That means that the selected value is excluded within the query. The value of the input field is for browsing only, the value is not used as query parameter.



Figure 4.6: Snippet of search parameter selection. (1) Property name. (2) Selection aid input field simplifies browsing within the list box (3). (3) List box containing all available values for the correlated property. Selected value can be inverted by tagging exclude check box(4).

Show All

The *Show All* mechanism allows presentation of the whole content of the database table **Mutein** only, without specifying any other options. This list view is created automatically for all entities which are tagged as **Manageable**. Attributes of the entity which are tagged as **ReportView** are selected by default for data representation. A detailed description of the list view is provided in section 4.4.3.

Search By Wildtype

This search mechanism allows listing of all muteins which share the same common wildtype protein. To perform a wildtype search it is mandatory to select a wildtype protein from the provided list box (2). The input field next to the list box (1), enables fast and easy browsing within the box entries see sec. 4.4.2.

The result is presented using the default list-view interface, see sec. 4.4.3.

Search By Inhibitor

Figure 4.8 shows the search mask for mutein search by their inhibitor molecule. The main search parameter is the inhibitor molecule, which is mandatory for performing an inhibitor search. It can be selected by browsing through all existing inhibitor molecules provided in the select box (2), (see sec. 4.4.2).

Figure 4.7: Search for muteins by wildtype proteins. (1) Selecting aid input field. (2) Select one menu box containing all available wildtype protein names. Selection of a wildtype name is mandatory for performing a query.

Figure 4.8: Illustration of search mutein by inhibitor molecule. (1) Selecting aid input field simplifies browsing within the correlated list box (2). The query can be refined by selecting wildtype (5) or mutein (6) name. Selected values can be excluded by tagging the correlated check box (3). For performing a search the selection of an inhibitor molecule is mandatory.

The search can be refined by defining two more search parameters. The result can be limited to special muteins (4) or wildtype proteins (5).

Search By Name

The search by name mechanism can be seen as an *universal* search mask, see fig. 4.9. At least one of the seven parameters, mutein name, wildtype name, substrate, EC-number, organism, tissue and expression host, has to be selected for performing a search. By specifying every parameter a very detailed search query can be built. The selection of mutein-, wildtype name, substrate, organism, tissue and expression host adhere to the basic selecting schema described in section 4.4.2. All available values for a particular parameter are listed in the respective select box (2, 4, 5, 7-9). The selection can be inverted by tagging the *exclude* check box right to the value box.

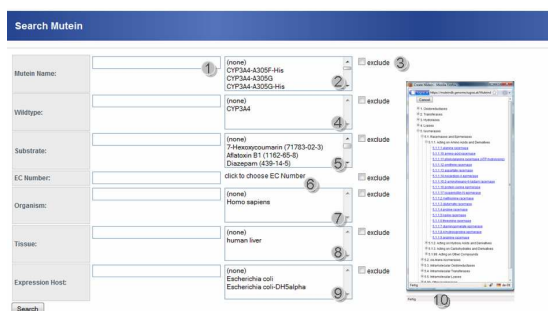


Figure 4.9: (1) Selection aid input field simplifies browsing in the corresponding select box (2). A detailed query can be built by specifying mutein name (2), wildtype name (4), substrate molecule name (5), organism (7), tissue (8) and expression host (9). Additionally an EC-number (6) can be selected from a popup EC-number tree (10). Chosen list box values can be excluded by tagging the correlated check box (3).

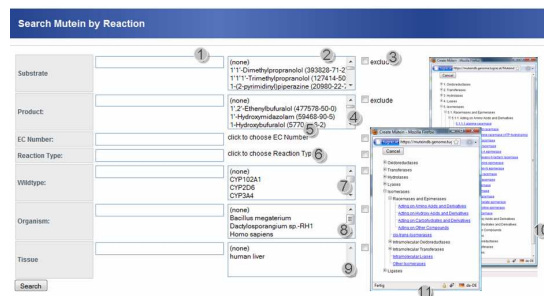


Figure 4.10: Illustration of search mutein by reaction web interface. (1) Selection aid input field, simplifies browsing to the correlated select list box (2). For performing a query at least a substrate (2) or a product (4) or an EC-number (5) or a reaction type (6) has to be specified. Additionally the search can be refined by selecting wildtype name (7), organism (8) or tissue (9). Popup windows (10)(11) provide selection aid for ec-number, reaction type respectively

The input field for the EC-number parameter is read only. The EC-number can be selected by clicking on a given link (5). An enzymatic class tree (10) is opened in a new window from which out the EC-number can be selected.

Additionally *wildcard* search functionality is supported for mutein name, substrate molecule name, and expression host name. In case of wildcard search the input field of the parameter need to contain at least one wildcard "*" symbol and total length has to exceed three characters. Otherwise the content of the input field of the parameter is discarded.

Substructure Search

With the integrated JME Molecular Editor (see sec. 3.5.6) into the application it is possible to search for molecules which contain a particular substructure pattern. Figure 4.11 illustrates the integrated applet with an arbitrary drawn structure (4) and the correlated SMILES representation (5). Usage of the applet is very simple and dose not require any manuals or other instructions. Structures are drawn using predefined atoms (2), bond types and shapes (3). The representation of the structure search result is illustrated and discussed in 4.4.3.

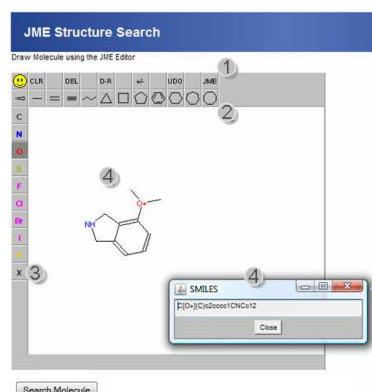


Figure 4.11: Illustration of substructure search web interface. (1) Main menu containing enabled features and base functions. (2)(3) Structure, molecule and bond shapes. (4) Drawing area containing a pattern created using the available shapes. (5) Smiles representation of the current applet content.

Search By Reaction

The search mutens by reaction data supports looking for mutens which catalyse the conversion of a particular substrate into a specific product. It is also possible to specify product or substrate parameter independently from each other. Furthermore the search can be performed by the selection of an EC-number (5) or a reaction type (6). The input fields for EC-number and reaction type are read only. Both values are set by choosing a value from the respective value tree presentation (10, 11). The search can be refined by specifying values for wildtype (7), organism (8) and tissue (9). Additionally wildcard search functionality is supported for product and substrate molecule name. In case of wildcard search the input field of the parameter need to contain at least one wildcard "*" symbol and total length has to exceed three characters. Otherwise the content of the input field of the parameter is discarded.

4.4.3 Search Result Presentation

The Search result presentation distinguishes between two different views. Results of *Show All*, *Search by Wildtype* and *Search by Inhibitor* use the default muten list view. Search results of *Search Muten* and *Search by Reaction* are presented using the *Muten Specific Activity Presentation* view.

Default Presentation

The default result representation, see fig. 4.12, is based on the automatically generated `muten-list` view. It provides the basic features implemented by the `ReportBean` mechanism. Every view created by a `ReportBean` contains a header (1) which comprises the page title (2) and the query (3) and edit display settings (4) feature. The number of overall found items is shown in (5). The data is divided into ranges of the selected page size (7), and tabular listed (6). Using the *Edit Display Settings* feature the displayed components of the table can be modified. The integrated paging mechanism (8) enables easy result browsing. Additionally it is possible to access a certain page directly, using the "go to page" mechanism (9). Authorised users are able to use the *Edit* (10), *Delete* (11) or the *New* feature.

Muten Specific Activity Presentation

The *Muten Specific Activity* presentation view, is based on the default presentation view and was manually created for the result presentation in case of *Search Muten* and *Search by Reaction*.

It comprises the same base `ReportBean` mechanisms already discussed within the *Default Presentation*, fig. 4.13 (1)-(11).

In addition to the default muten properties, the table also shows catalysis and activity related values, red box (12) fig. 4.13. For each search result entry (muten), the activity for the correlated catalysis with the highest specific activity is displayed. Further activities related to a particular catalysis are provided via (15). Figure 4.14 shows the *Muten Specific Activity Presentation* with expanded activity properties. The number of additional available activity entries for a muten entry is shown in

Name	Wildtype	Gene Bank ID	Go to page
1. CYP3A4			
2. CYP3A4L210A	CYP3A4	POSSA	
3. CYP3A4L211A	CYP3A4		
4. CYP3A4R023A	CYP3A4		
5. CYP3A4F213A	CYP3A4		
6. CYP3A4Q214A	CYP3A4		
7. CYP3A4F215A	CYP3A4		
8. CYP3A4L216A	CYP3A4		
9. CYP3A4M07g	CYP3A4		
10. CYP3A4Q214AHg	CYP3A4		
11. CYP3A4F3044Hg	CYP3A4		
12. CYP3A4A305G	CYP3A4		
13. CYP3A4A305VHg	CYP3A4		
14. CYP3A4L306AHg	CYP3A4		
15. CYP3A4L211FHg	CYP3A4		
16. CYP3A4Q214EHg	CYP3A4		
17. CYP3A4L1102HEM4	CYP3A4		
18. CYP3A4A305FHg	CYP3A4		
19. CYP3A4A305G	CYP3A4		

Figure 4.12: Illustration of the default search result presentation created by the Report Bean. (1) Header containing page title (2), and features (3)(4). (5) Number of overall found items. (6) Tabular listing of the search result. (7) Page size selecting option. (8) Paging bar. (10) edit, (11) delete option for an entry. (12) Add new entry option.

Wildtype	Name	Reaction Type	Substrate	Product	Km	Activity Value	Activity Unit	# Activities
CYP3A4	CYP3A4L210A	Hydroxylation	Progesterone	Ethyl-Hydroxyl	8.7		pmol/min/mg	16
CYP3A4	CYP3A4H6-T4	Hydroxylation	Testosterone	Ethyl-Hydroxyl	50.0		pmol/min/mg	37
CYP3A4	CYP3A4-F3044	Hydroxylation	Progesterone	Ethyl-Hydroxyl	22.8		pmol/min/mg	10
CYP3A4	CYP3A4-F3004-I	Hydroxylation	Progesterone	Ethyl-Hydroxyl	16.4		pmol/min/mg	10
CYP3A4	CYP3A4L2119E	Hydroxylation	Testosterone	Ethyl-Hydroxyl	137.0	44.0	pmol/min/mg	8
CYP3A4	CYP3A4A305G	S-oxidation	RPPI 100541	S-Sulfid-RPPI	16.74		pmol/min/mg	2
CYP3A4	CYP3A4A370V	S-oxidation	RPPI 100541	S-Sulfid-RPPI	20.98		pmol/min/mg	2
CYP3A4	CYP3A4Q214E	S-oxidation	RPPI 100541	R-Sulfid-RPPI	10.42		pmol/min/mg	2
CYP3A4	CYP3A4F106	Hydroxylation	Testosterone	Ethyl-Hydroxyl	8.79		pmol/min/mg	4
CYP3A4	CYP3A4-F3044	S-oxidation	RPPI 100541	S-Sulfid-RPPI	14.53		pmol/min/mg	2
CYP3A4	CYP3A4-G4802	Hydroxylation	Testosterone	Ethyl-Hydroxyl	25.2		pmol/min/mg	6
CYP3A4	CYP3A4L102L	Hydroxylation	Testosterone	Ethyl-Hydroxyl	14.9		pmol/min/mg	4
CYP3A4	CYP3A4G208	Hydroxylation	Testosterone	Ethyl-Hydroxyl	29.0		pmol/min/mg	4
CYP3A4	CYP3A4Q314	S-oxidation	RPPI 100541	S-Sulfid-RPPI	16.74		pmol/min/mg	2
CYP3A4	CYP3A4L030V	Hydroxylation	Testosterone	Ethyl-Hydroxyl	23.8		pmol/min/mg	6
CYP3A4	CYP3A4L120P	Hydroxylation	Testosterone	Ethyl-Hydroxyl	27.4		pmol/min/mg	4
CYP3A4	CYP3A4L211V	S-oxidation	RPPI 100541	S-Sulfid-RPPI	9.03		pmol/min/mg	2
CYP3A4	CYP3A4L478F	S-oxidation	RPPI 100541	S-Sulfid-RPPI	11.8		pmol/min/mg	2
CYP3A4	CYP3A4L473T	Hydroxylation	Testosterone	Ethyl-Hydroxyl	17.2		pmol/min/mg	4

Figure 4.13: Illustration of mutecin specific activity search result presentation. (1)-(11) Default ReportBean mechanisms, see fig. 4.12. (12) Additional added catalysis and activity properties columns. (13) All related activities are represented by the one with the highest specific activity value. (14) Number of additional available activity entries. (15) Load additional activities for the corresponding row.

column #Activities (1). By clicking on the plus symbol (2) the number of available activities is loaded into the current data model, red box (5). For sub elements the number of activities is set to zero and the fold out (2) option is disabled. Furthermore line-index and wildtype is not specified. Values can be collapsed by clicking the minus symbol (3).

Substructure Search Result Presentation

The result of the sub structure search is a list of molecules which contain the target pattern, illustrated in fig. 4.15. On the top of the result list the target pattern (1) is repeated. The result table contains molecule name (3) and identification number (2) for a positive match. The molecule ID can be either a Chemical Abstract Service (CAS) number or a PubChem Compound ID (CID). In any case the CID (4) is displayed for each result. The CID is linked to the corresponding compound entry in PubChem. Information about 2D molecule structure is provided by moving mouse cursor over the PubChem symbol.

4.4.4 Mutecin Data Presentation

Wildtype and mutecin names in default or mutecin specific activity result list view are linked to the respective detailed view. Clicking on the wildtype or mutecin name, will open the view containing the detailed enzyme properties, see fig. 4.16. Also IDs of other external databases are linked to the corresponding entry. All fields are *read only* in presentation mode. The enzyme data view is divided into the following five sections:

Basic Data

Presentation of the basic data related to current enzyme is divided into two sections, see fig. 4.17. Section *Basic Data* (1), shows basic information concerning the enzyme like source organism, tissue and molecular weight. *Create Information* (2), provides

Search Mutein Result List

Wildtype	Name	Reaction Type	Substrate	Product	KM	Activity Value	Activity Unit	# Activities
1	CYP3A4_CYP3A4-L215A	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	9.7	pmol/min/mg	10
2	CYP3A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	40.0	pmol/min/mg	37
3	CYP3A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	22.8	pmol/min/mg	10
4	CYP3A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	15.6	pmol/min/mg	10
5	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	15.0	pmol/min/mg	0
6	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	15.5	pmol/min/mg	0
7	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	0.3	pmol/min/mg	0
8	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	1.7	pmol/min/mg	0
9	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	0.1	pmol/min/mg	0
10	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	0.4	pmol/min/mg	0
11	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	0.3	pmol/min/mg	0
12	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	1.2	pmol/min/mg	0
13	CYP3A4-T208A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	16.9	pmol/min/mg	18
14	CYP3A4	Hydroxylation	Progesterone	Etoha-Hydroxyfen	100.0	44.0	pmol/min/mg	8
15	CYP3A4	S-oxidation	RPR 105541	S-Sulfoxid RPFIC	16.74	16.74	pmol/min/mg	2
16	CYP3A4	S-oxidation	RPR 105541	S-Sulfoxid RPFIC	20.99	20.99	pmol/min/mg	2
17	CYP3A4	S-oxidation	RPR 105541	S-Sulfoxid RPFIC	19.42	19.42	pmol/min/mg	2
18	CYP3A4	S-oxidation	RPR 105541	S-Sulfoxid RPFIC	9.79	9.79	pmol/min/mg	4
19	CYP3A4	S-oxidation	RPR 105541	S-Sulfoxid RPFIC	14.53	14.53	pmol/min/mg	2

JME Structure Search - Result

CAS number	name	related CID	PubChem structure view
84-97-9	Paroxetine		
20627-44-6	Paroxetine-S-sulphoxide		
3240-48-0	N-Desmethylparoxetine		
15532-75-9	1-(3-sulfonamethylphenyl)pyrrolidine		
58-40-2	Phenazine		
2056-20-7	N-Desmethylphenazine		
100569-03-4	8-Fluorenyl		
65277-42-1	Ketocanazole		
50-52-2	Thioindazole		
5088-33-0	Mesindazole		
14759-06-9	Sulfindazole		
7776-45-8	Thioindazole-S-Sulfoxide		
10308-32-6	N-Desmethylthioindazole	114276	
65027-34-3	Quindoline	4815	
65277-42-1	Ketocanazole	456201	
439-14-6	Diazepam	3016	
846-50-4	3-Hydroxydiazepam	5391	
50-49-7	Imipramine	3096	

Figure 4.14: Illustration of mutein specific activity search result presentation with extended activities details (5). (1) Number of collapsed activities for an enzyme entry. (2) Clicking on the plus symbol loads further activities for the selected entry into the current model. (3) Unloads sub elements. (4) Sub element rows do not contain row number or wildtype name.

Figure 4.15: (1) Repetition of search pattern. The result table comprises molecule ID (2), CAS or CID, molecule name (3), related CID (4), and a link to PubChem structure view (5).

information about registration date and user. The *Edit* button (3) is rendered for authorised users only. Section 4.4.5 describes the different supported edit modes.

Mutein - CYP3A4-A305F-His

Basic Data | Properties | Inhibition | Substrates | Sequence

Basic Data

Name: CYP3A4-A305F-His

Wildtype: CYP3A4

Organism: Homo sapiens

Kingdom: Mammalian

Tissue: human liver

Localization: micrososome (ER)

Length[Aa]: 503

Mol Weight[g/mol]: 56387.16

Num Mutations: 2

Gen Bank Id:

Uni Prot Id:

Swiss Prot Id:

Pdb Id:

Brenda Id:

Create Information

Create Date: 18.01.2009

Create User: Weinhandl

Contact:

Update Date:

Update User:

Cancel Edit

Mutein - CYP3A4-A305F-His

Basic Data | Properties | Inhibition | Substrates | Sequence

pH Range

pH Max [°C]:

pH Min [°C]:

pH Temp [°C]:

pH Opt Value:

pH Buffer:

pH Graph:

pH Graph Type:

pH Comment:

temp Range

Temp Max:

Temp Min:

Temp pH [°C]:

Temp opt Value:

Temp Buffer:

Temp Graph:

Temp Graph Type:

Temp Comment:

Storage Stability

Stability 1h:

Stability 4h:

Stability 20h:

Edit

Figure 4.16: Basic data is divided into two main sections: Basic Data and Create Information. Reference IDs, e.g. GenBankId, are linked to the correlated external entry. In case of muteins a link to the related wildtype protein is provided by clicking on the name. Permitted users are able to use *Edit* (3) option within the view.

Figure 4.17: Illustration of the preferences tab within the detailed enzyme view. Section (1) shows information about pH-Range, (2) about temp-Range and section (3) provides stability details.

Properties

The *Properties* tab, see fig. 4.17, illustrates enzyme properties divided into three sections. Section *pH Range* (1) outlines available information about pH-value range. Additional information about temperature value range is shown in the second section (2). Stability behaviour of the enzyme is provided in *Stability* section (3).

Inhibition

The *Inhibition* tab, see fig. 4.18, is only enabled when inhibitors have been entered for the mtuein entry. Within the tab a tabular listing (1) of all related inhibitor molecules and their effects on the enzyme is provided. Additional features like *Edit* (2) and *Add New* (3) are enabled for special user groups.

Sequence

The *Sequence* tab focuses on the presentation of sequence related information, see fig. 4.19. For wildtype proteins the sequence is presented in block view (1) and also in line view (2). Within the line view presentation all known mutations, of the wildtype sequence, are highlighted and linked to the corresponding mutein entry.

The *Block View* section, (1) in fig. 4.19, always shows the block view presentation of the selected enzyme. In case of wildtype the wildtype sequence, in case of mutein the modified sequence is displayed where mutated positions in the sequence are highlighted.

Name	Molecule Number	Ki Value	Ki Type	Temp	Method	Buffer	Solvent	Comments
1 Hydrocodamine	550-10-7	18.4	IC50	37.0	Fluorescence	potassium phosph		
2 Phenibutol	98040-97-6	4.1	Ki	37.0	HPLC	100 mM potassium		
3 Phenibutol	98040-97-6	4.1	Ki	37.0	HPLC	100 mM potassium		
4 CS 526	313272-10-7	6.0	IC50		Fluorescence	unknown		
5 HR40-47108	248919-64-4	3.8	IC50		Fluorescence	unknown		
6 AZD 0805	248281-68-7	9.0	IC50		Fluorescence	unknown		
7 SO-20089	78091-98-6	9.0	IC50		Fluorescence	unknown		
8 Sololol	3930-20-9			37.0	HPLC	100 mM potassium		
9 Sololol	3930-20-9	100.0	IC50	37.0	HPLC	100 mM TrisHCl		
10 Fluoxetine	54910-89-3	17.0	IC50	37.0	HPLC	100 mM TrisHCl		
11 Clozapine	439-14-2	100.0	IC50	37.0	HPLC	100 mM TrisHCl		
12 HR-60436	220835-95-6	400.0	IC50	37.0	HPLC	50 mM potassium		

Figure 4.18: Illustration of inhibition tab within the detailed enzyme view. The tab provides a tabular listing (1) of related inhibitor molecules and their effects on the enzyme. For permitted users the edit (2) and the add new (3) feature are enabled.

Name	Molecule Number	Ki Value	Ki Type	Temp	Method	Buffer	Solvent	Comments
1 Hydrocodamine	550-10-7	18.4	IC50	37.0	Fluorescence	potassium phosph		
2 Phenibutol	98040-97-6	4.1	Ki	37.0	HPLC	100 mM potassium		
3 Phenibutol	98040-97-6	4.1	Ki	37.0	HPLC	100 mM potassium		
4 CS 526	313272-10-7	6.0	IC50		Fluorescence	unknown		
5 HR40-47108	248919-64-4	3.8	IC50		Fluorescence	unknown		
6 AZD 0805	248281-68-7	9.0	IC50		Fluorescence	unknown		
7 SO-20089	78091-98-6	9.0	IC50		Fluorescence	unknown		
8 Sololol	3930-20-9			37.0	HPLC	100 mM potassium		
9 Sololol	3930-20-9	100.0	IC50	37.0	HPLC	100 mM TrisHCl		
10 Fluoxetine	54910-89-3	17.0	IC50	37.0	HPLC	100 mM TrisHCl		
11 Clozapine	439-14-2	100.0	IC50	37.0	HPLC	100 mM TrisHCl		
12 HR-60436	220835-95-6	400.0	IC50	37.0	HPLC	50 mM potassium		

Figure 4.19: Illustration of sequence information presentation for a mutein entry. Blue section numbers (1)(2) are only part of the wildtype sequence view. The mutein sequence view comprises all sections shown within the view, (1)-(6)

The sequence tab for a mutein sequence comprises six different sections, see fig. 4.19. All known mutations for the current mutein are listed at the top of the view (3). Modified positions are highlighted in the block view of the sequence in section (1). *Comments* (4), is only rendered in case a value is set for the field. At the bottom of the view a line-up of wildtype and mutein sequence is shown. In the *Signal Sequence* area (5)(6) respectively, related signal sequences are listed according to their (N- or C-terminal) location.

Substrate

Figure 4.20 illustrates catalysis and activity data presentation. All activities for a single catalysis are grouped together, (red box (1)). For each substrate catalysed by the enzyme the correlated reaction (2) and all resulting activities (5)(8) are listed. Entitled personnel are able to edit reaction (3), as well as activity (7) data and also add new catalyses (4) to the current enzyme. In case of mutein entries also the reference activity of the wildtype protein is rendered for each activity (6). Furthermore

information about *expression host*, *co-factor*, *co-protein* and *literature reference* are displayed within each activity section.

Figure 4.20: Illustration of the substrate tab within the detailed enzyme view. The tab includes all information about the specific activity(ies) measured for a particular substrate. The information is displayed in divisions of catalysis entries (1). For each catalysis (2) all related activities are listed (5)(8). Within the mutein substrate view also the reference activity for the corresponding wildtype is rendered (6).

4.4.5 Edit Interface

The following sections illustrate the web interfaces for editing the different resources. The *Edit* feature is enabled for authorised users only and can be accessed via the respective *view* interface. Not all properties of a particular resource can be changed. Mandatory fields are surrounded by a coloured border. Already set values are displayed in the respective input fields.

The following edit web interfaces are based on the automated generated edit views for a particular entity by the ReportBean. The views have been modified manually according to the needs of the application.

Basic Data

The basic data edit interface comprises the same values like the respective view. In section *Basic Data* (1), fig. 4.21, all fields which are not coloured grey (5) can be edited, whereby fields with a coloured border (3) have to contain values. For properties like, organism, kingdom, tissue and localisation it is also possible to add a new value to the drop down menu (4), which can be chosen afterwards. In section *Create Information* (2), only modification of the contact property is enabled.

Properties

The properties editing interface enables modification of *pH Range* (1), *Temp Range* (2), and the respective *Graph Information* (3). Graphical representation for temper-

Figure 4.21: Illustration of basic data edit view. (1) Section for basic data. (2) Section for user information. (3) Example for mandatory field. (4) Extension mechanism for the drop down menu. (5) Example for fields which are not free for change.

Figure 4.22: (1) section for pH Range data. (2) Section for temp Range data. (3) Section for graph information with upload option (6). (4) Section for determine or modify storage stability. (5) Signals option for extending the drop down values.

Figure 4.23: Illustration of inhibition edit view. (1) Molecule data read only, while other inhibition properties can be modified. (2) Extension option for drop down values.

ature and pH values can be uploaded (6). Furthermore the storage behaviour (4) of the selected enzyme can be set or modified. All the properties data is optional and modifiable. For buffer data, extending the existing selection is provided (5).

Inhibition

Parameters of existing inhibition entries can be modified. Figure 4.23 shows the inhibition edit interface. Modifications are restricted to property values. Primary inhibition data (1), inhibitor molecule name and identification number, is read only. Extensions to the dropdown menus of buffer, method and solvent can be made using the plus symbol (2).

Catalysis

Enzyme-catalysed reaction data can be modified partially. Within the catalysis edit interface authorised users are allowed to modify reaction molecule names, see fig. 4.24. Furthermore they are able to create a new reaction or delete the current one. Deletion of the catalysis will cause deletion of all activities related to it. Whereas substrate, product name can be modified molecule ID, EC-number and reaction type are read only.

Activity

Activity data of an enzyme-catalysed reaction can be modified via the activity edit interface. The view is divided into five main sections, see fig. 4.25. First section

contains main activity data. Second (2) and third (3) section comprises mutein, wildtype properties respectively. In the fourth (4) and fifth (5) section, related co-factors and co-proteins to the current activity are listed. Number of co-factors or co-proteins can be extended by adding further values via the according functions, (7)(8). Authorised users are also allowed to add new activity to the current catalysis or delete the active one.

Figure 4.24: (1) Substrate, product names are mandatory, but modifiable. (2) Product, substrate ID, EC-number and reaction type are read only. (3) Inactive data dictionary function. (4) Enables the creation of a new catalysis for the current enzyme. (5) Deletes the current enzyme and all related activity entries.

Figure 4.25: (1) Basic activity data. (2) Mutein activity properties. (3) Wildtype protein activity properties. (4) Listing of co-factors interacting to the activity. (5) Listing of co-proteins related to the activity. (7)(8) Add further co-factors, co-protein mechanism, respectively. (9) Create new activity for the current reaction. (9) Delete the current activity.

4.4.6 Data Import Interface

The data import web interface, allows Microsoft Excel File upload for automatic data import. A simple upload zone allows file system browsing for selecting an import source. The import starts when *Start Import* is pressed.

After the import has finished, a detailed report with or without errors is displayed. The import report is divided into three sections whereas only status and messages section are always displayed. The status section (1) contains the import status. It can either be positive, no errors occurred during the whole import process, or it can be negative due to serious problems during the value tree building or the database transaction (see sec. 4.3.1). Figure 4.26 gives an example for a positive feedback. The status messages (1) reports a positive end of the import process. A summary of the elements created during the import is listed in the message section (2). For elements which were added to the global DD detailed information via moving over the component (3). Figure 4.27 outlines a negative import feedback. The status section (1) informs about the failed import. The warning section (2) comprises events which occurred during the import, but do not cause serious problems. The focus of the negative feedback is on the error message section (3). In most cases it is a listing



Figure 4.26: Illustration of a positive data import feedback report. (1) Status section: gives information about successful or failing import. (2) Message section: summary of the import. (3) Listing of created buffer elements.

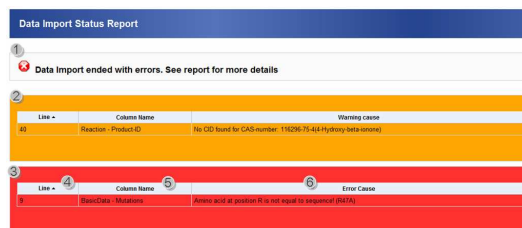


Figure 4.27: Illustration of a negative data import feedback report. (1) Status section: gives information about successful or failed import. (2) Warning section: contains messages which did not affect the import in a serious way. (3) Message section: detailed error report. (4) Line in which error occurred. (5) Column in which malformed value is contained. (6) Error description.

of errors which occurred during the value tree building phase because of invalid or missing cell values. It comprises line (4), column number (5) and the error message text (6).

Chapter 5

Discussion

The main goal of this thesis was the development of a management application for mutein data. Core requirements of the application were simple mechanism for data collecting, retrieval, maintenance, analysis and presentation. Under aspects of model driven software architecture, a three tiered Java Enterprise web application was implemented. MuteinDB 2.0 model was created by deriving it from the IGB AndroMDA3 Template Project. Based on this PSM all needed extensions were modelled using UML. Finally the code generation framework AndroMDA was used to translate the PSM into executable Java code. Choice of further used technologies depends on the AndroMDA3 Template Project standards. The core of the application is formed by an Oracle database at the persistence back end and a JBoss application server at the business tier. Business logic is encapsulated in EJB3. The web interface of the application is based on JavaServer Faces in combination with JBoss Seam.

5.1 Software Technologies

Software development using model driven aspects allow the implementation of interoperable, scalable, reusable high quality applications with reduced time and effort. Whereas the learning effort at the beginning is quite high, programmers benefit during implementation process from automated generation of entity, session and message driven beans code. Further data access objects that abstract and decouple database access to relational and object oriented databases as well as a service locator which provides abstraction and caching of initial context creation and Java Naming and Directory Interface (JNDI) lookups, are created. This mechanism allows programmers to focus on the business logic implementation of the application. The AndroMDA3 Template Project permits new applications to make use of already implemented base components like sharing mechanism, mail service, file upload mechanism, data dictionary and reporting mechanism. These benefits are in contrast to the platform independence approach of MDA because tailor-made stereotypes do not satisfy PIM requirements. The use of the AndroMDA3 Template Project results in restriction to the project standards and technologies. Nevertheless, the used Java Enterprise Edition is a well established platform for developing multitiered web applications. The main advantage of Java is its operating system independency, which allows Java applications to run on any platform, where a Java Runtime Environment is avail-

able. JEE which is used in the template project and this theses, provides a powerful set of APIs, that allow to reduce application complexity and improve performance. The newly introduced *annotations* within JEE make the use of XML deployment descriptors optional. The sophisticated EJB3 technology allows programmers to develop distributed, transactional, portable and secure Java Enterprise applications. Furthermore the new Java Persistence API provides object/relational mapping for managing relational data in beans and web components. Second core framework used aside from EJB3 is JavaServer Faces technology. It enables strict separation of presentation and behaviour of a web application. The most important architectural features of JSF are the abilities to manage component state, process component data, handling events and validating user input. Additionally to JSF the RichFaces component library was used. The well documented libraries, enable easy integration of AJAX capabilities into JSF web pages. AJAX components allow dynamic event handling, skinnability mechanisms aside to the whole JSF environment. Nevertheless JSF and EJB3 explore their full power in combination with JBoss Seam. It offers easy JSF - EJB3 integration and avoids the writing of repeating "glue" code between the two JEE components. Further benefits of JBoss Seam are the integrated Rich User Interface which offers Web 2.0 capabilities, validation mechanisms and annotations that reduce XML definitions. MuteinDB 2.0 makes use of the integrated AAS of the AndroMDA3 Template Project. It provides well structured and proven mechanism for user and resource management. Hence, simplified and secure permission control and user maintenance is guaranteed.

5.2 Application Features

Compared to other databases which offers mutein related information, MuteinDB 2.0 uniquely provides catalysis and kinetic information. The main benefit is the option to search for enzymes by using reaction related information. This mechanism simplifies retrieving enzyme as well as specific activity information related to a particular substrate that is catalysed during an enzymatic reaction. The well structured search result presentation focuses on the maximal specific activity of each enzyme. Additional, available activity data can be displayed on demand. Hence, extracting information relevant for researchers is now much easier. Related to the search result presentation using the provided report mechanism for manageable entities, some serious problems arose. By default the integrated report mechanism for manageable entities returns a list of entities which fulfil the query restrictions. In case of *multiplicity* between related entities each reference is considered as a match. As a consequence multiple entities with the same PK are contained in the result list. Because catalysis and activity properties can be specified in different search terms these parameters have to be taken into account. To fulfil the defined purpose of the result presentation the automatically generated report mechanism had to be replaced by an extended version, which was manually created. Therefore, the report mechanism was changed to return mutein PK and related catalysis and activity PK instead of mutein entities. Based on this result list made up of arrays of mutein, catalysis and activity PKs a hash table is created. This hash groups catalysis and activities according to their mutein relation. This enables easy removing of duplicate entries, fast and targeted sub element (catalysis, activity) retrieval and avoiding redundant database queries.

Unfortunately these modifications result in the loss of flexible and automated code generation for the affected entities.

Another improvement is the new substructure search option. It allows to find molecules which contain a certain structure pattern. Compounds found can then be used in a further *Search by Reaction* to retrieve detailed activity information. The integration of data import mechanisms for MS Excel files enables easy and fast data collecting. Data is persisted according to *the all or non principle*, within the same transaction. This guarantees, that only complete sets of enzymes are written to the database. In case of errors during writing to the database, the transaction is marked for rollback and the database is restored to the state before the transaction was started. The detailed import feedback outlines import statistics. In case of errors, line and column according to the import file and the root cause of the error are reported to the user. A successful data import ends with a detailed summary of the new items. This feedback mechanisms enable user integration throughout the import procedure. With a variety of integrity checks and validation mechanisms and the import of information from third party databases a high data quality can be reached. The application makes heavy use of the data dictionary, provided by the AndroMDA3 Template Project. Based on a predefined set of values for different domains, users are able to select a particular value or add a new one to a certain domain. This enables standardised and consistent naming of different attributes like buffers, solvents, co-factors and expression hosts. Based on uniform data input, querying performance and quality is increased.

5.3 Outlook

The application as it is allows protein, mutein data collecting from literature sources. It is currently not possible to register unpublished data. To enable the management of ongoing researches, the *Create Mutein* web interface could be adjusted to allow *private*, not published working sets. Entries defined as working sets could easily be marked with a Boolean *public or private* flag in database table. In course of this, extension report mechanism has to be adjusted to the effect that *working set data* is not considered. Additionally to this mechanism a related retrieval function for research data has to be developed as well. Since the result of ongoing researches will not produce a lot of data, an extension of the data import Excel template and the import mechanism will not be necessary.

MuteinDB 2.0 provides one- and two- dimensional structure presentation mechanisms for proteins/muteins. As physical configuration, three dimensional structure of the protein is essential for its selectivity and hence its functionality, it would be a major improvement to provide tertiary structure presentation as well. Therefore, the according information could be imported from a third party database for example Protein Mutant Database using the according entry PDBID. If the predicted three dimensional structure information is available for the particular enzyme, structure information could be exported to a format supported by a 3D structure viewer like JMol.

All known mutations of a particular wildtype sequence are highlighted and illus-

trated within the detailed sequence enzyme view web interface. From this description it is not possible to extrapolate to the likeliness of mutations for a particular position. Hence, a graphical illustration of the known mutation distribution at a position would be more meaningful.

To enhance mutein literature research an automated data mining system could be integrated into the application. This features would support to specify at least one key word which is searched within the abstract of a literature database like PubMed. As result e.g. PubMedIDs are listed that contain at least one keyword in order of relevance. The main record could be made available by clicking on the ID.

During the data import, terms not specified in the data dictionary, within a certain domain, are automatically added. Even though a predefined set of values is used within the Excel file for several components, this method is not aware of considering "new" entries caused by typing errors. To lessen this problem the degree of similarity of the current value to existing values could be calculated using the *Levenshtein Distance*³¹ [60]. Distances above a certain threshold are treated as typing-error candidates. To handle such "false positive" data dictionary entries, a intermediate step, comprising user interaction, before the storage in the database would be necessary. Import is halted for required user confirmations, and continued afterwards.

³¹Metric algorithm for determining the difference between character sequences

Bibliography

- [1] Halwachs B. *Developersguide AndroMDA3*. Institute for Genomics and Bioinformatics, Graz University of Technology, Petersgasse 14, 8010 Graz, Austria, 2009.
- [2] Crick F. *Central dogma of molecular biology*. Nature, 1970, 8:227-561.
- [3] IUPAC-IUB Commission on Biochemical Nomenclature. *A one-letter notation for amino acid sequences, tentative rules*. J Biol Chem, 1968, 243(13):3557-3559.
- [4] Watson JD, Berry A. *DNA: The Secret of Life*. 1st edition, New York, NY:USA, Alfred A. Knopf Inc.; 2003.
- [5] Cell Division, Reproduction, and DNA. *Codonsun*.
<http://www.ck12.org/ck12/images?id=293240/>
May 5th, 2010
- [6] Comparison of activity energy with and without a biocatalyst. *Diagram activation activity comparison*.
<http://simpert.com/resources/Enzyme+01.jpg/>
May 5th, 2010
- [7] Fischer E. *Einfluss der Configuration auf die Wirkung der Enzyme*. Ber Dtsch Chem Ges, 1894, 27(3):2985-2993.
- [8] Building the enzym substrate complex.
<http://simpert.com/resources/Enzyme+02.jpg/>
May 5th, 2010
- [9] Schomburg I, Chang A, Hofmann O, Ebeling C, Ehrentreich F, Schomburg D. *Brenda: a resource for enzyme data and metabolic information*. Trends Biochem Sci, 2002, 27(1):54-56.
- [10] Kawabata T, Ota M, Nishikawa K. *The protein mutant database*. Nucleic Acids Res, 1999, 27(1):355-7.
- [11] Bava KA, Gromiha MM, Uedaira H, Kitajima K, Sarai A. *Protherm, version 4.0: thermodynamic database for proteins and mutants*. Nucleic Acids Research, 2004, 32:D120-121.
- [12] Structural prediction for protein folding utility system.
<http://bioinformatics.eas.asu.edu/sprouts.html/>
May 5th, 2010

- [13] Lonquety M, Lacroix Z, Papandreou N, Chomilier J. *Sprouts: a database for the evaluation of protein stability upon point mutation*. Nucleic Acids Res, 2009, 37:D374-9.
- [14] Preissner S, Kroll K, Dunkel M, Senger C, Goldsobe G, Kuzman D, Guenther S, Winnerburg R, Schroeder M, Reissner R. *SuperCYP: a comprehensive database on Cytochrome P450 enzymes including a tool for analysis of CYP-drug interactions*. Nucleic Acids Res, 2010, 37:D237-D243.
- [15] Guggemos M. *Mutein Datenbank*. Master's thesis, Institute for Molecular Biotechnology, Graz University of Technology, Petersgasse 14, 8010 Graz, Austria, 2006.
- [16] UML - Unified Modeling Language.
<http://www.uml.org/>
 May 5th, 2010
- [17] UML - What is UML.
http://www.omg.org/gettingstarted/wha__is__uml.htm
 May 5th, 2010
- [18] MDA - Model Driven Architecture.
<http://www.omg.org/mda/>
 May 5th, 2010
- [19] JEE - Java Enterprise Edition.
<http://java.sun.com/j2ee/overview.html/>
 May 5th, 2010
- [20] JEE Application Model.
<http://java.sun.com/javaee/5/docs/tutorial/doc/bnaay.html/>
 May 5th, 2010
- [21] Enterprise Java Beans 3.0.
<http://java.sun.com/products/ejb/>
 May 5th, 2010
- [22] Bergsten H. *JavaServer Faces*. 1st edition, Sebastopol, CA:USA, O'Reilly Media Inc.; 2004.
- [23] JBoss Rich Faces.
<http://www.jboss.org/richfaces/>
 May 5th, 2010
- [24] Yuan M, Heute T. *JBoss Seam: Simplicity and power beyond Java EE*. 1st edition, Upper Saddle River, NJ:USA, Prentice Hall International; 2007.
- [25] AndroMDA.
<http://www.andromda.org/>
 May 5th, 2010
- [26] Truskaller T. *Data integration into a gene expression database*. Master's thesis, Institute for Genomics and Bioinformatics, Graz University of Technology, Petersgasse 14, 8010 Graz, Austria, 2003.

- [27] Sayers E, Wheeler D. *Building Customized Data Pipelines Using the Entrez Programming Utilities (eUtils)*. 1st edition, Bethesda, MD:USA, NCBI; 2004.
- [28] Bolton EE, Wang Y, Thiessen PA, Bryant SH. *PubChem: integrated platform of small molecules and biological activities*. *Annu Rep Comput Chem*, 2008 4:217-241.
- [29] PubChem - Compound Home.
<http://www.ncbi.nlm.nih.gov/sites/entrez?db=pccompound>.
May 5th, 2010
- [30] Power User Gateway - A Service of PubChem.
<http://pubchem.ncbi.nlm.nih.gov/pug/pughelp.html/>
May 5th, 2010
- [31] CrossRef.
<http://www.crossref.org/help/>
May 5th, 2010
- [32] Wang J. *Digital Object Identifiers and Their Use in Libraries*. *Serials Review.*, 2007, 33(3):161-164.
- [33] Eclipse - Software Development Environment.
<http://www.eclipse.org/>
May 5th, 2010
- [34] Magice Draw UML.
<http://www.magicdraw.com/>
May 5th, 2010
- [35] Apache Maven.
<http://maven.apache.org/>
May 5th, 2010
- [36] Apache Maven Central Repository.
<http://repo1.maven.org/maven2/>
May 5th, 2010
- [37] Oracle SQL Developer.
http://www.oracle.com/technology/products/database/sql_developer/index.html/
May 5th, 2010
- [38] CAS REGISTRY and CAS registry numbers.
<http://www.cas.org/expertise/cascontent/registry/regsys.html>.
May 5th, 2010
- [39] Check digit verification of CAS registry numbers.
<http://www.cas.org/expertise/cascontent/registry/checkdig.html>.
May 5th, 2010
- [40] Webb E, International Union of Biochemistry, and Molecular Biology. *Enzyme nomenclature 1992 : Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the nomenclature and classification of enzymes*. revised edition 1984, New York, NY:USA, Academic Press; 1992.

- [41] Weininger D. *Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules.* J Chem Inf Comput Sci, 1988, 28:31-36.
- [42] Wiswesser WJ. *A Line Formular Chemical Notation.* 1st edition, New York, NY:USA, Crowell Co; 1954.
- [43] Daylight Chemical Information Systems Inc. *Daylight Theory Manual.* Daylight Chemical Information Systems Inc., 120 Vantis - Suite 550 - Aliso Viejo, CA 92656, USA, 2008.
- [44] Weininger D, Weininger A, Weininger JL. *Smiles 2: algorithm for generation of unique smiles notation.* J Chem Inf Comput Sci, 1989, 29:97-101.
- [45] Morgen HL. *The generation of a unique machine description of chemical structures - a technique developed at chemical abstract service.* J Chem Inf Comput Sci, 1965, 5:107-113.
- [46] Leach AR, Gillet VJ. *An introduction to chemoinformatics.* 1st edition, Dordrecht:Netherlands, Springer Netherlands; 2007.
- [47] Apache - POI.
<http://poi.apache/>
May 5th, 2010
- [48] Zeller D. *Design and development of a user management system for molecular biology.* Master's thesis, Institute for Genomics and Bioinformatics, Graz University of Technology, Petersgasse 14, 8010 Graz, Austria, 2003.
- [49] Maurer M, Molitor R, Sturn A, Hartler J, Hackl H, Stocker G, Prokesch A, Scheideler M, Trajanoski Z. *Mars: microarray analysis, retrieval, and storage system.* BMC Bioinformatics, 2005, 6(1):101-113.
- [50] Steinbeck C, Han Y, Kuhn S, Horlacher O, Luttmann E, Willighagen E. *The Chemistry Development Kit (CDK): an open-source java library for chemo- and bioinformatics.* J Chem Inf Comput Sci, 2003, 43:493-500.
- [51] Fechner U. *Frequently asked questions.* CDK News, 2004, 1(2):6-7.
- [52] *CDK Java Class Documentation.*
<http://pele.farmbio.uu.se/nightly-1.2.3/cdk-javadoc-1.2.4/index.html>
May 5th, 2010
- [53] Daylight, Chemical Information Systems Inc. *Fingerprints - screening and similarity.*
<http://www.daylight.com/dayhtml/doc/theory/theory.finger.html>.
May 5th, 2010
- [54] Brown R, Martin Y. *Use of structure-activity data to compare structure-based clustering methods and descriptors for us in compound selection.* J Chem Inf Comput Sci, 1996, 36:572-584.
- [55] Tonnelier C, Jauffret Ph, Hanser Th, Kaufmann G. *Machine Learning of Generic Reactions: 3. an efficient algorithm for maximal common substructure determination.* Tetrahedron Comput Methodol, 1990, 3:351-358.

- [56] JDOM.
<http://www.jdom.org/>.
May 5th, 2010
- [57] overLIB.
<http://www.bosrup.com/web/overlib/>.
May 5th, 2010
- [58] Ertl P. JME Molecular Editor.
<http://www.molinspiration.com/jme/>.
May 5th, 2010
- [59] Pan K, Zhang, Sun H, Luo L. *An implementation of substructure search in chemical database management system*. International Multi-symposiums on Computer and Computational Sciences, IMSCCS'08, 18-20 October 2008, Shanghai, China, 2008:203-206.
- [60] Levenshtein VI. *Binary codes capable of correcting deletions, insertions and reversals*. Sov Phys Dok, 1966, 10(8):707-710.

Appendix A

JME Licence Agreement

"The JMEMolecularEditor is a computer software to which Novartis AG has the exclusive copyright. Installation of the JMEMolecularEditor by any user shall only be permitted when user accepts and agrees to give proper credit to Novartis AG, including naming the author of the JMEMolecularEditor, Peter Ertl, and Novartis AG within the used HTML page.

User shall install and/or use the JMEMolecularEditor only to draw and edit chemical structures as input to his chemical information and modeling systems, and user shall in no event decompile or otherwise disassemble the JMEMolecularEditor.

Installation and use of the JMEMolecularEditor shall only be permitted when user accepts and agrees

- that Novartis does not warrant fitness of the JMEMolecularEditor for any general or special purpose whatsoever, and*
- that Novartis shall in no event be liable to anybody for any damage, loss or personal injury which may result directly or indirectly from the installation, use, deletion or any other handling of the JMEMolecularEditor, and*
- that any user of the JMEMolecularEditor shall indemnify and hold Novartis harmless from any and all claims by third parties related to the user's installation, use or other handling of the JMEMolecularEditor.*

" [58]

Appendix B

PUG conversation example

The following listings provide an example for the retrieval of SMILES for a set of PubChem Compound Ids (CID). The conversation starts by sending the XML request, see listing B.1, to the CGI script.

```
1 <PCT-Data>
2   <PCT-Data_input >
3     <PCT-InputData>
4       <PCT-InputData_download >
5         <PCT-Download >
6           <PCT-Download_uids >
7             <PCT-QueryUids >
8               <PCT-QueryUids_ids >
9                 <PCT-ID-List >
10                  <PCT-ID-List_db>pccompound</PCT-ID-List_db >
11                  <PCT-ID-List_uids >
12                    <PCT-ID-List_uids_E>970</PCT-ID-List_uids_E >
13                    <PCT-ID-List_uids_E>12704</PCT-ID-List_uids_E >
14                  </PCT-ID-List_uids >
15                </PCT-ID-List >
16              </PCT-QueryUids_ids >
17            </PCT-QueryUids >
18          </PCT-Download_uids >
19          <PCT-Download_format value="smiles" />
20          <PCT-Download_compression value="none" />
21        </PCT-Download >
22      </PCT-InputData_download >
23    </PCT-InputData >
24  </PCT-Data_input >
25 </PCT-Data >
```

Listing B.1: XML request for retrieving SMILES representation for 3 given CIDs

PUG is responding (see code listing B.2) to the request in listing B.2 with an response containing the waiting ID 402936103567975582 and the submit state <PCT-Status value="success"/>.

```
1 <PCT-Data>
2   <PCT-Data_output>
3     <PCT-OutputData>
4       <PCT-OutputData_status>
5         <PCT-Status-Message>
6           <PCT-Status-Message_status>
7             <PCT-Status value="success"/>
8           </PCT-Status-Message_status>
9         </PCT-Status-Message>
10        </PCT-OutputData_status>
11       <PCT-OutputData_output>
12         <PCT-OutputData_output_waiting>
13           <PCT-Waiting>
14             <PCT-Waiting_reqid>402936103567975582</PCT-Waiting_reqid>
15           </PCT-Waiting>
16         </PCT-OutputData_output_waiting>
17       </PCT-OutputData_output>
18     </PCT-OutputData>
19   </PCT-Data_output>
20 </PCT-Data>
```

Listing B.2: PUG XML response containing the request ID

Using the waiting ID a request (see listing B.3) is polled periodically to PUG for retrieving the result.

```
1 <PCT-Data>
2   <PCT-Data_input>
3     <PCT-InputData>
4       <PCT-InputData_request>
5         <PCT-Request>
6           <PCT-Request_reqid>402936103567975582</PCT-Request_reqid>
7           <PCT-Request_type value="status"/>
8         </PCT-Request>
9       </PCT-InputData_request>
10      </PCT-InputData>
11    </PCT-Data_input>
12  </PCT-Data>
```

Listing B.3: Programmatic poll request to PUG

Finally the result of the task is returned by the PUG (see listing B.4).

```
1 <PCT-Data>
2   <PCT-Data_output>
3     <PCT-OutputData>
4       <PCT-OutputData_status>
5         <PCT-Status-Message>
6           <PCT-Status-Message_status>
7             <PCT-Status value="success"/>
8           </PCT-Status-Message_status>
9         </PCT-Status-Message>
10      </PCT-OutputData_status>
11     <PCT-OutputData_output>
12       <PCT-OutputData_output_structure>
13         <PCT-Structure>
14           <PCT-Structure_structure>
15             <PCT-Structure_structure_string>
16               970: C(C(=O)C(=O)O)C(=O)O
17             </PCT-Structure_structure_string>
18             <PCT-Structure_structure_string>
19               12704: CCN(CC)C(=S)SC
20             </PCT-Structure_structure_string>
21           </PCT-Structure_structure>
22           <PCT-Structure_format>
23             <PCT-StructureFormat value="smiles"/>
24           </PCT-Structure_format>
25         </PCT-Structure>
26       </PCT-OutputData_output_structure>
27     </PCT-OutputData_output>
28   </PCT-OutputData>
29 </PCT-Data_output>
30 </PCT-Data>
```

Listing B.4: PUG XML response containing the result

Appendix C

Usermanagement integration example

The code snippet in listing C.1 illustrates the usage of the integrated user management. The current access level is fetched from the user management server for the correlated resource and stored to `hasPermission` variable. Using the `haspermission` value the rendering, and the access control can be managed.

```
13 ....
14
15 <ui:composition template="/faces/layout/layout.xhtml">
16   <ui:define name="title">
17     <c:set var="title" value="Change Password" scope="request"/>
18   </ui:define>
19   <ui:define name="headerBackgroundColor">
20     <c:set var="headerBackgroundColor" value="Blue" scope="page"/>
21   </ui:define>
22
23   <ui:define name="content">
24     <c:set var="hasPermission"
25       value="#{aas:permission('changePassword','R')}" />
26     <ajax:form id="changePswdForm" rendered="#{hasPermission}">
27       ....
28     </ajax:form>
29
30     <rich:spacer height="10px"/>
31     <rich:panel rendered="#{!hasPermission}">
32       <h:graphicImage url="/images/stop.png"
33         styleClass="StandardTableCellEven"/>
34       <h:outputText value="#{messages['error.rights.notSufficient']}"
35         styleClass="StandardTableCellEven"/>
36     </rich:panel>
37   </ui:define>
38 </ui:composition>
```

Listing C.1: Code snippet for `aas:permission` user management tag

Appendix D

UML Models

D.1 MuteinDB 1.0 UML Diagram

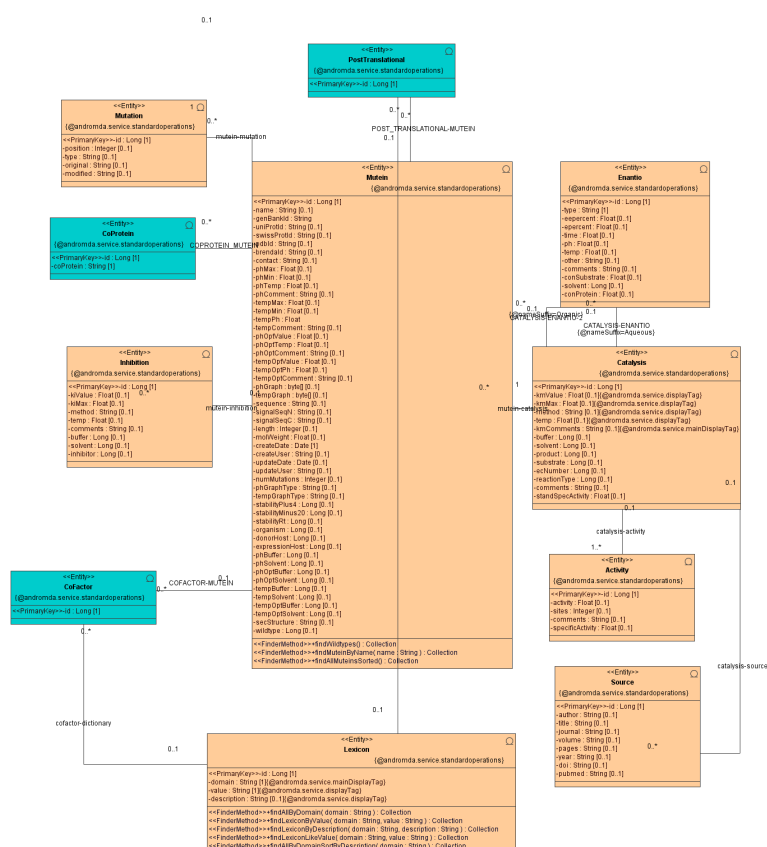


Figure D.1: Entity domain UML model showing database tables of MuteinDB 1.0 and their relations.

D.2 MuteinDB 2.0 UML Diagram

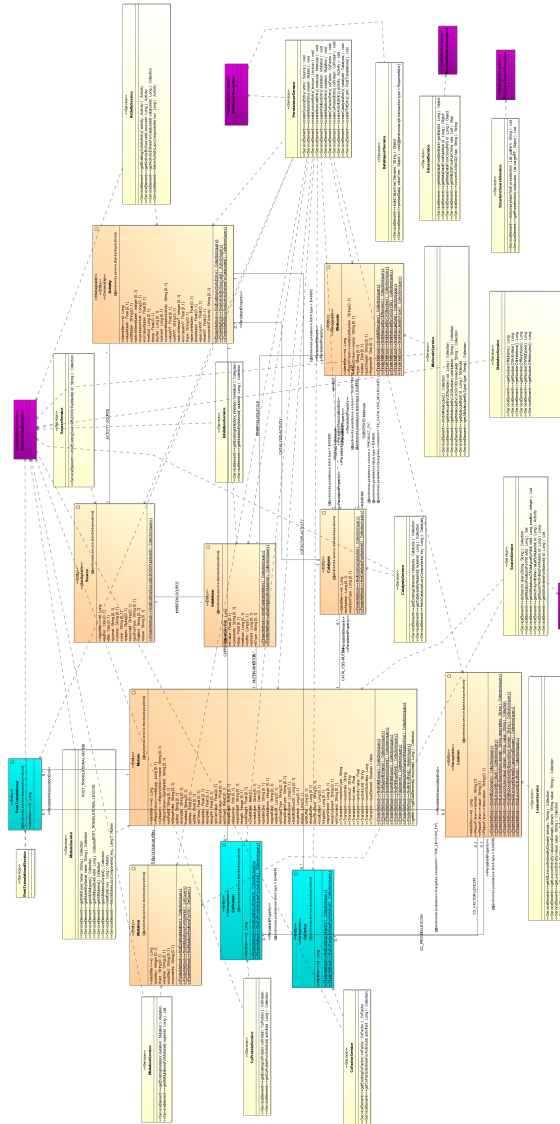


Figure D.2: Entity domain model of MuteinDB 2.0. The figure illustrates all MuteinDB 2.0 entities (tables)(orange/blue), services(yellow) and exception classes(pink) and their relations.

Appendix E

Meta Data Retrieval

E.1 XML Response for EFetch GenBankID request

Listing E.1 illustrates the XML response for wildtype protein Cytochrom P450 with GenBankID P08684. Primary sequence information and additionally organism name and sequence length are extracted and stored in database.

```
1 <?xml version="1.0"?>
2 <!DOCTYPE TSeqSet PUBLIC "-//NCBI//NCBI TSeq/EN" "http://www.ncbi.nlm.nih.gov/dtd/
   NCBI_TSeq.dtd">
3 <TSeqSet>
4 <TSeq>
5   <TSeq_seqtype value="protein"/>
6   <TSeq_gi>116241312</TSeq_gi>
7   <TSeq_accver>P08684.4</TSeq_accver>
8   <TSeq_taxid>9606</TSeq_taxid>
9   <TSeq_orgname>Homo sapiens</TSeq_orgname>
10  <TSeq_defline>RecName: Full=Cytochrome P450 3A4; AltName: Full=Quinine 3-
    monooxygenase; AltName: Full=CYP11A4; AltName: Full=Nifedipine oxidase;
    AltName: Full=Cytochrome P450 3A3; AltName: Full=CYP11A3; AltName: Full=
    Cytochrome P450 H1p; AltName: Full=Tauroch<gt;</TSeq_defline>
11  <TSeq_length>503</TSeq_length>
12  <TSeq_sequence>
13  MALIPDLAMETWLLLAVSLVLLYLYGTHSHGLFKKLGIPGPTPLPFLGNILSYHKGFCMFDMECHKKYGKVGWGFYDGGQPV
14  LAITPDMIKTVLVKECYSVFTNRRPFGPVGFMKSAISIAEDEEWKRLRSLSPFTFTSGKLEMPVPIAQYGDVLRNLR
15  EAETGKPVTLKDFGAYSMDVITSTSFVGNIDSLNPNQDPFVENTKLLRFDLDFLDFLFFLSITVFPFLIPILEVLNICVFP
16  EVTNFLRKSVKRMKESRLEDTKHRVDFLQMLMDSQNSKETESHKALSDLELVAQSIIFIFAGYETTSSVLSFIMYELATH
17  PDVQKQLQEEIDAVLPNKAPPTYDVTVLQMEYLDVVNETLRLFPFIAMRLERVCKKDVVEINGMFIPKGVVMMIPSYALHRDP
18  KYWTEPEKFLPERFSKKNKDNIDPYIYTPFGSGPRNCIGMRFALMNMKLALIRVLQNFSPKPKETQIPLKLSLGLLQPE
19  KPVVLKVESRDGTVSGA
20 </TSeq_sequence>
21 </TSeq>
22 </TSeqSet>
```

Listing E.1: eUtils XML response for GenBankID P08684

E.2 XML Response for CAS 2 CID Transformation

Listing E.2 shows the retrieved XML response for CAS 103-79-7 within the CAS to CID transformation. The result of the transformation is CID 7678 was found for CAS number 103-79-7

```
1 <eSearchResult>
2   <Count>1</Count>
3   <RetMax>1</RetMax>
4   <RetStart>0</RetStart>
5   <IdList>
6     <Id>7678</Id>
7   </IdList>
8   <TranslationSet/>
9   <TranslationStack>
10    <TermSet>
11      <Term>103-79-7[All Fields]</Term>
12      <Field>All Fields</Field>
13      <Count>1</Count>
14      <Explode>Y</Explode>
15    </TermSet>
16    <OP>GROUP</OP>
17  </TranslationStack>
18  <QueryTranslation>103-79-7[All Fields]</QueryTranslation>
19 </eSearchResult>
```

Listing E.2: eUtils URL for transforming CAS (103-79-7) into CID

E.3 XML Respons for PMID meta data query

Listing E.3 illustrated the ESummary DocSum response for PubMedID 10191269. This result is used to extract all meta data which is needed for database storage.

```
1 <eSummaryResult >
2   <DocSum >
3     <Id>10191269</Id>
4     <Item Name="PubDate" Type="Date">1999 Apr 15</Item>
5     <Item Name="EPubDate" Type="Date"/>
6     <Item Name="Source" Type="String">Biochem J</Item>
7
8     <Item Name="AuthorList" Type="List">
9       <Item Name="Author" Type="String">Noble MA</Item>
10      <Item Name="Author" Type="String">Miles CS</Item>
11      <Item Name="Author" Type="String">Chapman SK</Item>
12      <Item Name="Author" Type="String">Lysek DA</Item>
13      <Item Name="Author" Type="String">MacKay AC</Item>
14      <Item Name="Author" Type="String">Reid GA</Item>
15      <Item Name="Author" Type="String">Hanzlik RP</Item>
16      <Item Name="Author" Type="String">Munro AW</Item>
17    </Item>
18    <Item Name="LastAuthor" Type="String">Munro AW</Item>
19
20    <Item Name="Title" Type="String">
21      Roles of key active-site residues in flavocytochrome P450 BM3.
22    </Item>
23    <Item Name="Volume" Type="String">339 ( Pt 2)</Item>
24    <Item Name="Issue" Type="String"/>
25    <Item Name="Pages" Type="String">371-9</Item>
26    <Item Name="LangList" Type="List">
27      <Item Name="Lang" Type="String">English</Item>
28    </Item>
29    <Item Name="NlmUniqueID" Type="String">2984726R</Item>
30    <Item Name="ISSN" Type="String">0264-6021</Item>
31    <Item Name="ESSN" Type="String">1470-8728</Item>
32
33    <Item Name="PubTypeList" Type="List">
34      <Item Name="PubType" Type="String">Journal Article</Item>
35    </Item>
36    <Item Name="RecordStatus" Type="String">PubMed - indexed for MEDLINE</Item>
37    <Item Name="PubStatus" Type="String">ppublish</Item>
38
39    <Item Name="ArticleIds" Type="List">
40      <Item Name="pubmed" Type="String">10191269</Item>
41      <Item Name="pmc" Type="String">PMC1220167</Item>
42      <Item Name="pmcid" Type="String">pmc-id: PMC1220167;</Item>
43    </Item>
44
45    <Item Name="History" Type="List">
46      <Item Name="pubmed" Type="Date">1999/04/07 00:00</Item>
47      <Item Name="medline" Type="Date">1999/04/07 00:01</Item>
48      <Item Name="entrez" Type="Date">1999/04/07 00:00</Item>
49    </Item>
50    <Item Name="References" Type="List"/>
51    <Item Name="HasAbstract" Type="Integer">1</Item>
52    <Item Name="PmcRefCount" Type="Integer">4</Item>
53    <Item Name="FullJournalName" Type="String">The Biochemical journal</Item>
54    <Item Name="ELocationID" Type="String"/>
55    <Item Name="S0" Type="String">1999 Apr 15;339 ( Pt 2):371-9</Item>
56  </DocSum >
57 </eSummaryResult >
```

Listing E.3: ESummary DocSum response for PMID 10191269

E.4 CrossRef Meta Data Service Response

Listing E.4 illustrates the XML response of the CMS for DOI 10.1002/adsc.200505069. This result is used to extract all meta data which is needed for database storage.

```
1 <doi_record xmlns="http://www.crossref.org/xschema/1.0" owner="10.1002" timestamp
2   ="2007-08-02 09:25:45.0">
3   <crossref>
4     <journal>
5       <journal_metadata language="en">
6         <full_title>Advanced Synthesis & Catalysis</full_title>
7         <abbrev_title>Advanced Synthesis & Catalysis</abbrev_title>
8         <issn media_type="print">1615-4150</issn>
9         <issn media_type="electronic">1615-4169</issn>
10        <doi_data>
11          <doi>10.1002/(ISSN)1615-4169</doi>
12          <resource>
13            http://doi.wiley.com/10.1002/%28ISSN%291615-4169
14          </resource>
15        </doi_data>
16      </journal_metadata>
17      <journal_issue>
18        <publication_date media_type="print">
19          <month>06</month>
20          <year>2005</year>
21        </publication_date>
22        <journal_volume>
23          <volume>347</volume>
24        </journal_volume>
25        <issue>7-8</issue>
26        <doi_data>
27          <doi>10.1002/adsc.v347:7/8</doi>
28          <resource>
29            http://doi.wiley.com/10.1002/adsc.v347:7/8
30          </resource>
31        </doi_data>
32      </journal_issue>
33      <journal_article publication_type="full_text">
34        <titles>
35          <title>Converting Phenylacetone Monooxygenase into
36            Phenylcyclohexanone Monooxygenase by Rational Design:
37            Towards Practical Baeyer-Villiger Monooxygenases
38          </title>
39        </titles>
40        <contributors>
41          <person_name contributor_role="author" sequence="first">
42            <given_name>Marco</given_name>
43            <surname>Bocola</surname>
44          </person_name>
45          <person_name contributor_role="author"
46            sequence="additional">
47            <given_name>Frank</given_name>
48            <surname>Schulz</surname>
49          </person_name>
50          <person_name contributor_role="author"
51            sequence="additional">
52            <given_name>François</given_name>
53            <surname>Leca</surname>
54          </person_name>
55          <person_name contributor_role="author"
56            sequence="additional">
57            <given_name>Andreas</given_name>
58            <surname>Vogel</surname>
59          </person_name>
60          <person_name contributor_role="author"
61            sequence="additional">
62            <given_name>Marco?W.</given_name>
63            <surname>Fraaije</surname>
64          </person_name>
65          <person_name contributor_role="author"
66            sequence="additional">
```

```
66         <given_name>Manfred?T.</given_name>
67         <surname>Reetz</surname>
68     </person_name>
69 </contributors>
70 <publication_date media_type="print">
71     <month>06</month>
72     <year>2005</year>
73 </publication_date>
74 <pages>
75     <first_page>979</first_page>
76     <last_page>986</last_page>
77 </pages>
78 <doi_data>
79     <doi>10.1002/adsc.200505069</doi>
80     <resource>
81         http://doi.wiley.com/10.1002/adsc.200505069
82     </resource>
83 </doi_data>
84 <citation_list/>
85 </journal_article>
86 </journal>
87 </crossref>
88 </doi_record>
```

Listing E.4: CMS response for DOI 10.1002/adsc.200505069

Appendix F

Data Import MS Excel Template

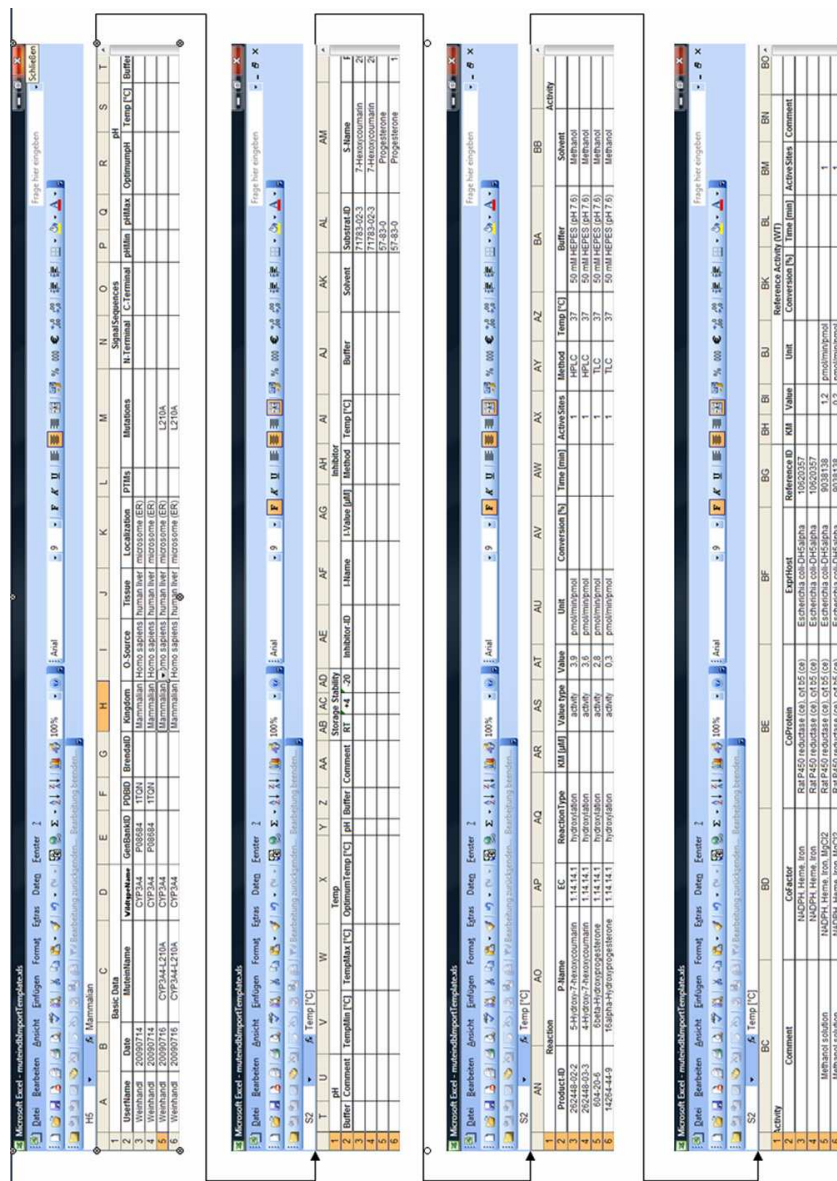


Figure F.1: Illustration of all group (line 1), and column (line 2) names of the MS Excel template file

Appendix G

Data Import Guidelines

GENERAL RULES
DATA IMPORT MUTEIN-DB

V2010

Generally it's essential to register as much information as possible to obtain a meaningful and reliable data base. Mostly literature contains data for more than one mutein and therefore we created some excel files to facilitate data entry (available at gerhard.thallinger@tugraz.at, each enzyme has its own file!); these files are finally imported to mutein data base by Gerhard Thallinger.

Before registering new data, please check, if they are really absent in the database to avoid double-entries!!

EXCEL FILE:

Do not change the template (only add data)

Delete all empty rows between data rows!

The Excel file is separated in different areas (first row):

- Basic data
- Signal sequence, pH, Temperature, Stability
- Inhibitor
- Reaction
- Activity
- Reference activity

	A	B	C	D	E	F	G	H	I	J	K
1			Basic Data								
2	UserName	Date	MuteinName	WildtypeName	GenBankID	PDBID	BrendalD	Kingdom	O-Source	Tissue	Localization
3	Weinhandl	20090824		CYP3A4	P08684	1TQN		Mammalian	Homo sapiens	human liver	microsome (ER)
4	Weinhandl	20090824	CYP3A4-L293P	CYP3A4				Mammalian	Homo sapiens	human liver	microsome (ER)
5	Weinhandl	20090824		CYP51	P01369	2KGW		Yeast	Saccharomyces cerevisiae-AH22		peroxisome
6	Weinhandl	20090824	CYP51-M445T	CYP51				Yeast	Saccharomyces cerevisiae-AH22		peroxisome
7	Braun	20100119		CYP3A4	P08684	1TQN		Mammalian	Homo sapiens	human liver	microsome (ER)
8	Braun	20100119	CYP3A4-L211F/D214E	CYP3A4				Mammalian	Homo sapiens	human liver	microsome (ER)
9	Braun	20100120		CYP3A4	P08684	1TQN		Mammalian	Homo sapiens	human liver	microsome (ER)
10	Braun	20100120	CYP3A4.53x	CYP3A4				Mammalian	Homo sapiens	human liver	microsome (ER)
11	Braun	20100120		CYP3A4	P08684	1TQN		Mammalian	Homo sapiens	human liver	microsome (ER)
12	Braun	20100120	CYP3A4.53-1	CYP3A4				Mammalian	Homo sapiens	human liver	microsome (ER)
13											

Fig.1.: Area “Basic data”

Following issues should be considered to obtain a consistent appearance:

For each mutant/ WT activity please start a new row!

Each excel-file should start with a WT entry (required for the data import procedure)

BASIC DATA:

- Username: = Last Name
- Date: Date of entry (format: YYYYMMDD e.g.: 20091024)
- MuteinName: consists of WT-Name, hyphen, mutated position (e.g.: CYP2D6-R440H); if the mutein contains more mutations, use slash (e.g. CYP2D6-R440H/S486T); naturally existing mutants have own names (e.g. CYP2D6.31); Sometimes authors worked with muteins containing more than three mutations and gave them individual names, in this case assume that name out of literature. (if necessary add WT-Name in front e.g.: literature 132-10 → CYP2D6-132-10);
-

Example	Entry
one Mutation (R440H)	CYP2D6-R440H
more Mutations (R440H and S486T)	CYP2D6-R440H/S486T
naturally occurring mutant (due to polymorphism)	CYP2D6.31
individual names (132-10)	CYP2D6-132-10
TAG's (HIS-TAG)	CYP2D6-HIS-TAG
Fusion Protein (CYP2D6 fused with CPR)	CYP2D6-CPR

Tab.1.: Entry of mutein names

If a Wildtype is entered, leave “MuteinName” empty

- WildtypeName: for muteins always specify corresponding WT (e.g. CYP2D6)
- GenbankID (WT: mandatory because of sequence import, muteins: only if explicit for the mutein)

- PDB-ID, BRENDA ID (WT: if available, muteins: only if explicit for the mutein)
if there are several PDB entries the first should be the one containing cofactors but no ligands (natural form), add the others separated by comma and space (e.g. 1rg5, 6fs9)
- Kingdom = organism kingdom (use dropdown menu e.g. mammalian)
- O-Source = name of the origin organism (“complete name”, eg. Homo sapiens, if a strain designation is known: “complete name” hyphen “strain designation” e.g. Saccharomyces cerevisiae-AH22)
- Tissue, Localization = natural occurrence
(Tissue: e.g. humane liver, Localization: use dropdown menu e.g. microsomes (ER))
- PTMs = Post translational modifications of the enzyme (Multiple PTMS are separated by comma and space eg: phosphorylation, glycosylation)
- Mutations = mutated position: = **mandatory for muteins!**
Numbering: Sequence includes the start Methionine (M = Number 1)
(several mutations are listed using comma and space e.g. R44H, D101F; muteins, that contain one or more changes, separate by using coma and space e.g. R44H, CTAGHHHHHH, T97-W98insLQS)

Mutation	Example	Description
Substitution	R440H	R440 is substituted to H
Deletion* (del)	T97-C102del	T97 – C102 is deleted
	T97del	T at position 97 is deleted
Insertion* (ins)	T97-W98insLQS	Between T97 and W98 L, Q and S are inserted
TAG	CTAGHHHHHH NTAGHHHHHH	TAG containing 6 H (His) at the C-terminus / N-terminus
fusion (fus)	CfusHCPR NfusHCPR	fusion of the enzyme HCPR (humane cytochrome P450 reductase) at the C-terminus / N-terminus
truncation (tru)	Ctru10 Ntru10	10 AS are cut of at the C-terminus / N-terminus

Tab.2.: Entry of mutations

*Source: JT den Dunnen and SE Antonarakis, 2000, Human mutation 15: 7-12

SIGNAL SEQUENCE, PH, TEMPERATURE, STABILITY

- Signal Sequences: N- and C-terminal sequences (amino acids; e.g. PLLLLALV)
- pH: conditions of pH tests (buffer, temperature, pH min and max, resulting optimum pH)
- temperature: conditions of temperature tests (buffer, temp min and max, resulting optimum temperature) – unit of temperature is always °C
- Stability (e.g.stable or unstable)

INHIBITOR

- Inhibitor ID: (if unknown type unknown)
CAS-number (look at www.scifinder.at or <http://ctd.mdibl.org>); e.g. 56-54-2,
CID (look at <http://pubchem.ncbi.nlm.nih.gov/>); e.g. CID 1615
- I-Name = Inhibitor, e.g. Quinidine
- I-Value: Inhibitor Value in [μM] (always use the same unit!)
Ki or IC50 (IC50-Values always with the prefix “IC50” and “space” eg. IC50 35)
- Method: method of measurement to get I-value (e.g. HPLC)
- Reaction conditions → temperature, buffer: use dropdown menu, if the buffer is not there, add the buffer in the “dropdown-table” (e.g. 50 mM potassium phosphate (pH 7.2))
- Solvent: solvent for Inhibitor (e.g. Methanol, if unknown type “unknown”) for Solvent mixtures separate the solvents with a “slash” (e.g. Methanol/Water)
do not enter specific data like concentration etc.

For all Inhibitor entries the Substrate and Product, which were used to measure the Inhibitor values, have to be given in the Reaction section (only Substrate and Product Name and ID are needed, if they are not known write unknown as Name and leave the ID-column empty) Comments about the Inhibition can be given in the comment column in the Reaction section.

For all kind of molecules (inhibitors, substrates, products) a CAS number is mandatory! (If no CAS-number is available, use CID (PubChem CompoundID) with prefix “CID” and “space” e.g.: CID 1615, or 56-54-2. If no prefix is used it is treated as CAS, if no CAS and no CID is available type unknown)

REACTION AND ACTIVITY

- Substrate ID Product ID → (if unknown type unknown)
preferred CAS-number (look at www.scifinder.at or <http://ctd.mdibl.org>);
e.g. 54340-62-4,
alternatively CID (look at <http://pubchem.ncbi.nlm.nih.gov/>); e.g. CID 1615
- S-Name = Substrate, e.g. Bufuralol
- P-Name = Product, e.g. 1-Hydroxybufuralol

For molecules, that contain alpha, beta etc., please don't use special signs, write out the whole name (e.g. 16alpha-Hydroxyprogesterone)!

If products (or reaction types) are unknown, register them as “unknown” (field for CAS = empty, don't enter characters like “-“, “ND” etc.)

If the structure is given and just named with a number (e.g. M1, 2a,...), enter the chemical name, otherwise register them as “unknown”.

Although some conversions give several products in one reaction, nevertheless create an own row for each product.

If a Protein is inactive towards a certain substrate, create a normal entry (observed reaction, e.g. Substrate, Product, Unit etc.) and enter “0” for Activity Value.

- EC-Number
- Reaction type: use dropdown-menu, if the reaction type is not there, add the reaction type in the “dropdown-table” (e.g. hydroxylation, no position, except (N-demethylation, O-demethylation, etc.)

AT	AU	AV	AW	AX
Reaction				
S-Name	Product	P-Name	EC	ReactionType
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation
Bufuralol	57704-16-2	1-Hydroxybufuralol	1.14.14.1	1-hydroxylation
Bufuralol	57704-16-2	1-Hydroxybufuralol	1.14.14.1	1-hydroxylation
Bufuralol	57704-16-2	1-Hydroxybufuralol	1.14.14.1	1-hydroxylation
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation
Debrisoquine	59333-79-8	4-Hydroxydebrisoquine	1.14.14.1	4-hydroxylation
Debrisoquine	59333-79-8	4-Hydroxydebrisoquine	1.14.14.1	4-hydroxylation
Debrisoquine	59333-79-8	4-Hydroxydebrisoquine	1.14.14.1	4-hydroxylation
Debrisoquine	59333-79-8	4-Hydroxydebrisoquine	1.14.14.1	4-hydroxylation
Debrisoquine	59333-79-8	4-Hydroxydebrisoquine	1.14.14.1	4-hydroxylation
Dextromethorphan	125-73-5	Dextrorphan	1.14.14.1	O-demethylation

Fig.2.: Area “Reaction”

- Km: always use **μM!** (no special sign’s, e.g. “>”)
- ValueType: Type of measured Value (use dropdown menu e.g. activity)
- Value: (no special sign’s, e.g. >, “0” for inactive enzymes)
- Unit: use dropdown menu, for activity preferentially use the unit **pmol/min/pmol!!!!**; (to convert mg Protein to pmol we recommend to use http://www.molbiol.ru/eng/scripts/01_04.html for the calculation) if there is no activity per Protein given, also “nmol/min/mg total protein”, “nmol/min/g CDW” are allowed;
- enantiomeric excess in % e.e; E-Value has no Unit (empty field)

- Inactive enzymes: give the Valuetype, the Unit and value = 0
If value type is unknown, leave all three fields empty.
- Conversion % and Time:
only in combination with enantionmeric excess or E-value
- Active sites: number of active sites (for CYP2D6: 1)

AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI
KM [μM]	Value type	Value	Unit	Conversion [%]	Time [min]	ActiveSites	Method	Temp [°C]	Activity Buffer
	activity	0,15	pmol/min/pmol			1	TLC	37	50 mM HEPES buffer (pH 7.6)
	activity	0,3	pmol/min/pmol			1	TLC	37	50 mM HEPES buffer (pH 7.6)
	activity	0,12	pmol/min/pmol			1	TLC	37	50 mM HEPES buffer (pH 7.6)
	activity	0,12	pmol/min/pmol			1	TLC	37	50 mM HEPES buffer (pH 7.6)
						1	HPLC	37	100 mM potassium phosphate buffer (pH 7.4)
						1	HPLC	37	100 mM potassium phosphate buffer (pH 7.4)
135	activity	11	pmol/min/pmol			1	HPLC	37	50 mM Heps (pH 7.6)
129	activity	73	pmol/min/pmol			1	HPLC	37	50 mM Heps (pH 7.6)
35,9	activity	1,5	pmol/min/pmol			1	HPLC	37	100 mM potassium phosphate buffer (pH 7.4)
68,3	activity	0,5	pmol/min/pmol			1	HPLC	37	100 mM potassium phosphate buffer (pH 7.4)
57,2	activity	0,9	pmol/min/pmol			1	HPLC	37	100 mM potassium phosphate buffer (pH 7.4)
129,7	activity	0,5	pmol/min/pmol			1	HPLC	37	100 mM potassium phosphate buffer (pH 7.4)
34,2	activity	1,2	pmol/min/pmol			1	HPLC	37	100 mM potassium phosphate buffer (pH 7.4)

Fig.3.: Area „Activity“

- Reaction conditions → Method (e.g. HPLC), Temp.(always in °C), Buffer:
use dropdown menu, if the buffer is not there, add the buffer in the
“dropdown-table” (e.g. 50 mM potassium phosphate (pH 7.2))
- Solvent for Substrate (e.g. Methanol)
(if unknown type “unknown”, for Solvent mixtures separate the solvents
with a “slash” e.g. Methanol/Water, do not enter specific data like
concentration etc.)
- Comment: special information about the reaction, that can’t be registered
anywhere else (e.g. if values were estimated from graphics, special
reaction conditions,
- CoFactors and –proteins: present during the described reaction (e.g. for
CYP2D6: Cofactor = NADPH, Heme, Iron; Coprotein = Human P450
reductase → in parenthesis: if this protein is co-expressed (ce) or added to
reaction (atr)). (Multiple Cofactors – Proteins are separated by comma and

space

eg. NADPH, FAD)

- Expression host: organism, in which recombinant DNA was expressed use dropdown menu, if the host is not there, add the host in the “dropdown-table”
(“complete name” hyphen “strain designation”; e.g. Escherichia coli-JM109)
- Reference ID: please give information about the literature source; either PubmedID or DOI-ID should appear for each entry (**preferred ID = PubMed, DOI with prefix and “space” e.g. DOI 10.1248/jhs.50.503!**
why PubMed ID: a link to the PubMed entry will be created during the import process)

REFERENCE ACTIVITY (WT)

(same rules apply as described for REACTION AND ACTIVITY)

In a WT row the Reference Activity (WT) section is not needed.

Additionally as a reference for mutein data, each mutein entry should also include the WT activity data. (only enter if it was measured in the corresponding paper, do not enter WT activity from other sources)

- $K_m \rightarrow \mu\text{M}$
- Value: unit \rightarrow same as used for REACTION AND ACTIVITY
- Conversion % and Time:
only in combination with enantionmeric excess or E-value
- Active sites

Import will be successful, if all mandatory fields contain valid values. Look at Table 2 for more details.

		Wildtype	Mutein	Comments
Basic Data	UserName	mandatory	mandatory	
	Date	mandatory	mandatory	
	MuteinName	leave empty	mandatory	
	WildtypeName	mandatory	mandatory	
	GenBankID	mandatory		
	PDB-ID			
	Brenda-ID			
	Kingdom	mandatory	mandatory	
	O-Source	mandatory	mandatory	
	Tissue			
	Localization			
	PTMs			
	Mutations		mandatory	sequence includes start Methionine (Number = 1)
SignalSequences	N-Terminal			
	C-Terminal			
pHRange	pHMin			
	pHMax			
	OptimumpH			
	Temp [°C]			
	Buffer			
	Comment			
Temp	TempMin [°C]			
	TempMax [°C]			
	OptimumTemp [°C]			
	pH			
	Buffer			
	Comment			
Storage Stability	RT			
	+4			
	-20			
Inhibitor	Inhibitor-ID (CAS or CID)	mandatory	mandatory	In case of inhibition
	I-Name	mandatory	mandatory	In case of inhibition
	I-Value [µM]	mandatory	mandatory	In case of inhibition
	Method			
	Temp [°C]			
	Buffer			
	Solvent			
Reaction	Substrat-ID (CAS or CID)	mandatory	mandatory	„ unknown “ if unknown

	S-Name	mandatory	mandatory	
	Product-ID (CAS or CID)	mandatory	mandatory	
	P-Name	mandatory	mandatory	
	EC	mandatory	mandatory	
	ReactionType	mandatory	mandatory	„unknown“ if unknown
Activity	KM [μ M]			inactive value = 0
	Value Type	mandatory	mandatory	
	Value	mandatory	mandatory	
	Unit	mandatory	mandatory	
	Conversion %			
	Time [min]			
	ActiveSites	mandatory	mandatory	in most cases „1“
	Method			
	Temp			
	Buffer			
	Solvent			
	Comment			
	CoFactor			
	CoProtein			
	ExpreHost	mandatory	mandatory	
Reference ID (PubMedID, DOI)	mandatory	mandatory	Use PubMedID or DOI	
Reference Activity (WT)	KM [μ M]	empty		
	Value	empty	mandatory	
	Unit	empty	mandatory	
	Conversion [%]	empty		
	Time [min]	empty		
	ActiveSites	empty	mandatory	
	Comment	empty		

Please send your actual file to Gerhard Thallinger.

ENTRY IN MUTEIN-DATABASE

If you want to register just one or a few entries, it's easier to use the „Create“-application at <https://muteindb.genome.tugraz.at/Muteindb/>
(Please request accession data at Gerhard Thallinger)

The general entry rules are the same like for the Excel File, but the surface and the basic areas are a bit different:

1. **Create** → Choose, if you want to register Mutein or Wildtype data
2. **Basic information:** Give Name, IDs (as described in area „Basic data“ in the Excel file), choose a kingdom, Tissue, Localization, O-Source (= natural habitat) and the expression host in the drop down menu or create a new one („New button“) but please avoid double entries! Please add your user ID in „Availability/Contact“.
3. **Properties:** as described in the area pH, Temperature, Stability and Inhibitor of the Excel File; for new Inhibitors: „Add inhibitor“;
4. **Substrates:** choose or add Substrates and Products (don't forget CAS- or CID- numbers!), activity in pmol/min/pmol, enantiomeric excess (ee%), Km in μM ; choose or add „Cofactor“, „Coprotein“; or Expressionhost Literature source,
5. **Sequence;** area for mutations (AA mutation, codon mutation), signal sequences

Summary of entry format:

Topic	Format/unit	example
Date	year month day	20091024
Km	μM	-
Activity Units	[pmol/min/pmol] only if not otherwise possible [nmol/min/mg total Protein], [nmol/min/g CDW]	-
Buffer	mM buffer name (pH)	100 mM potassium phosphate(pH 7.4)
I-Value (Ki or IC50)	μM	53, IC50 53
Reference ID	PubMed ID (alternative = DOI with prefix)	16269134 DOI 10.1248/jhs.50.503
CAS	alternative = CID with prefix	33817-09-3 CID 1615
Expression Host	“complete name” hyphen “strain designation”	Escherichia coli-DH5alpha