INFORMATION SOCIETY TECHNOLOGIES
(IST)
PROGRAMME

OpenMolGRID

SPECIFICATION OF THE WORKFLOW FOR THE OVERALL MOLECULAR ENGINEERING PROCESS

Contract Reference:	**IST-2001-37238**
Document identifier:	**OpenMolGRID-4-D4.4-0119-0-1-WorkflowOverallProcess**
Date:	**19/04/2004**
Work package:	**WP 4: Grid Integration**
Partner:	**UT, UU, Negri, FZJ, CGX**
Lead Partner:	**FZJ**
Document status:	**APPROVED**
Classification:	**PUBLIC**
Deliverable identifier:	**D4.4**

Abstract: Description and Definition of the workflow elements for the process of molecular engineering

Delivery Slip

	Name	Partner	Date
From	M.Romberg	FZJ	08/04/2004
Verified by	M.Romberg	**FZJ**	19/04/2004
Approved by	G.H.F.Diercksen (TC) R.Ferenczi (QE)	OMC CGX	11/04/2004 16/04/2004

Document Log

Issue	Date	Comment	Author
0-0	08/04/2004		Mathilde Romberg
0-1	19/04/2004		Mathilde Romberg

Document Change Record

Issue	Item	Reason for Change
0-1		Input from UT and CGX

Files

Files in this section relate to actual storage locations on the BSCW server located at https://hermes.chem.ut.ee/bscw/bscw.cgi. The URL below describes the location on BSCW from the root OpenMolGRID directory

Software Products	User files / URL
Word 2000/XP	OpenMolGRID/Workpackage 4/Deliverables/ OpenMolGRID-4-D4.4-0119-0-1-WorkflowOverallProcess

Project information

Project acronym:	OpenMolGRID
Project full title:	Open Computing GRID for Molecular Science and Engineering
Proposal/Contract no.:	IST-2001-37238
European Commission:	
Project Officer:	Annalisa BOGLIOLO
Address:	European Commission - DG Information Society F2 - Grids for Complex Problem Solving B-1049 Brussels Belgium
Office	BU31 4/79
Phone:	+32 2 295 8131
Fax:	+32 2 299 1749
E-mail	annalisa.bogliolo@cec.eu.int
Project Coordinator:	Mathilde ROMBERG
Address:	Forschungszentrum Jülich GmbH ZAM D-52425 Jülich Germany
Phone:	+49 2461 61 3703
Fax:	+49 2461 61 6656
E-mail	m.romberg@fz-juelich.de

Contents

1. Introduction

1.1. Purpose and scope

This document describes the workflow the chemist follows when doing molecular engineering. It covers the steps of generating target structures and evaluating these structures to identify promising candidates. It contains the data flow and the control flow of the molecular engineering process as relevant for OpenMolGRID. Based on the workflows for Descriptor Calculation and Model Development already defined in [5]and [6] this document focuses on structure generation and property/activity evaluation.

1.2. Document Overview

OpenMolGRID's objective is to automatise the molecular engineering process in support of the molecular engineer. The process uses computer algorithms for the construction of molecular structures with the predefined chemical property or biological activity values. These algorithms help scientists to explore large chemical space in a cost effective way for finding potential candidates for new drugs, chemicals, or materials.

The molecular engineering process, described in [1], involves a number of important tasks that are carried out in different stages of the complicated workflow. The prerequisites for the molecular engineering process are good predictive models for the theoretical validation of potential candidate molecules. These models are developed using the data mining tools developed in WP2 and embedded in the OpenMolGRID system (see [5], [6]) The generation of new molecular structures is based on the predefined library of fragment structures. With this library, various structure generation algorithms can construct a huge number of candidate structures. These structure generation algorithm use fragment descriptors for a quick selection of candidate structures. The candidate structures are validated with previously developed predictive models and a small subset of molecules that match the target properties or activities are selected for the further investigation.

On the basis of the high-level flowchart for a typical drug design problem the workflow for the overall molecular engineering process is detailed. The workflow is specified at different views: Verbal, as data flow model, and as flowchart to describe all aspects of the process. The XML file with the exact input for the MetaPlugin (see [4]) is attached.

1.3. Document Structure

The document contains in Section 2 a general description of the drug design process as being seen by the OpenMolGRID project. Section 3 contains the different views of the workflow, Section 4 and Section 5 hold references and glossary, and Appendix 1 gives the XML representation of the workflow.

2. Drug Design

A realistic drug design scenario looks like the following: The task is to find a set of molecular structures with given target properties, where the activity (IC50) of HIV-1 protease inhibitors is higher than X, solubility is higher than Y, and carcinogenicity is lower than Z. The process flow is depicted in Figure 1. It shows three major sub-processes that need to be carried out to determine the molecular structures that exhibit the desired properties: (1) the data warehousing process (depicted by single line boxes), (2) the data mining process (double line boxes), and (3) the molecular engineering process (thick, single line boxes). Those processes that rely heavily on accessing and using computational services (high-performance, high-throughput) are highlighted by a shadow. The data warehousing part provides the data in form and quality the subsequent steps need as basis. The workflow to be looked at then starts with querying the data warehouse and calculating necessary descriptors for the model development (see [5]). The next steps are those relevant for developing appropriate statistical models which are going to be used within the actual molecular engineering process.

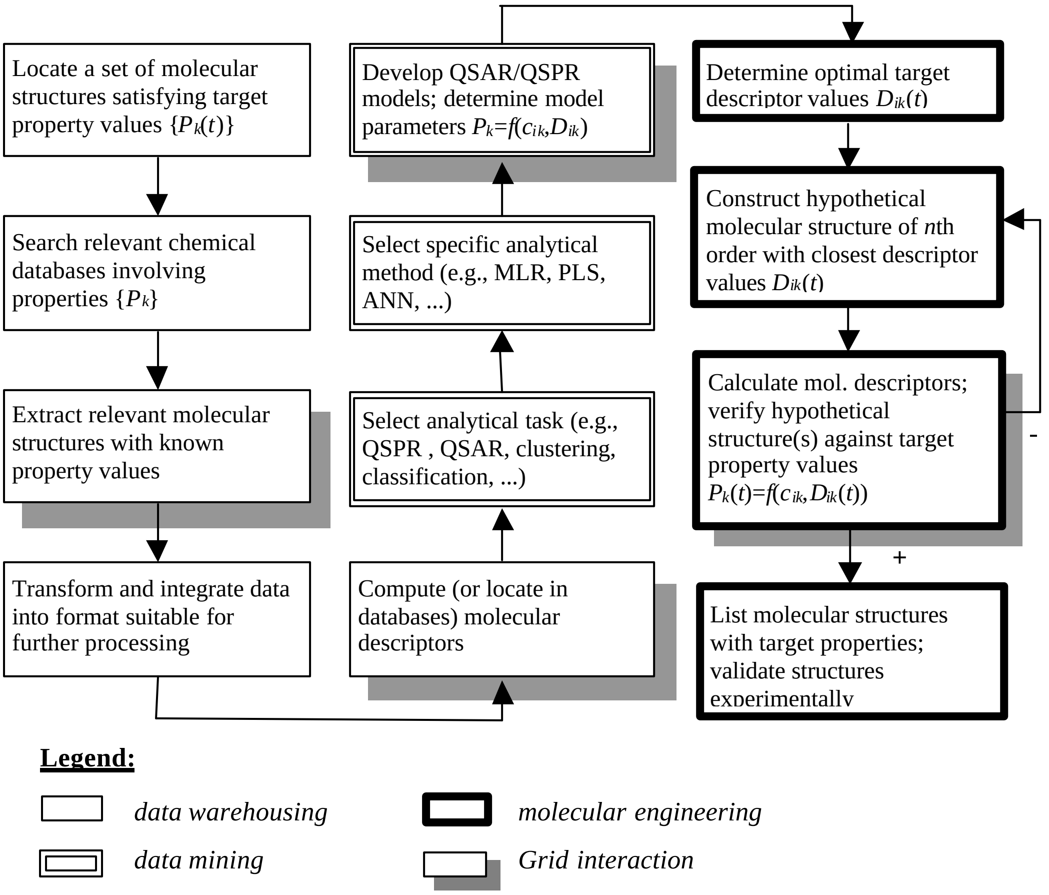
Locate a set of molecular structures satisfying target property values {$P_k(t)$}
Search relevant chemical databases involving properties {P_k}
Extract relevant molecular structures with known property values
Transform and integrate data into format suitable for further processing
Compute (or locate in databases) molecular descriptors
Select analytical task (e.g., QSPR , QSAR, clustering, classification, ...)
Select specific analytical method (e.g., MLR, PLS, ANN, ...)
Develop QSAR/QSPR models; determine model parameters $P_k=f(c_{ik},D_{ik})$
Determine optimal target descriptor values $D_{ik}(t)$
Construct hypothetical molecular structure of *n*th order with closest descriptor values $D_{ik}(t)$
Calculate mol. descriptors; verify hypothetical structure(s) against target property values $P_k(t)=f(c_{ik},D_{ik}(t))$
-
+
List molecular structures with target properties; validate structures experimentally
Legend:
data warehousing
molecular engineering
data mining
Grid interaction

Figure 1: Flowchart for typical drug design problem

3. Workflow Specification

The workflow for molecular engineering starts with developing suitable predictive models. Based on an available set of fragment structures it generates candidate structures and evaluates these using the predictive models to identify the most promising molecules which then would be synthesised and tested in the laboratory. In the following sections the workflow is given in detail.

3.1. Workflow Description

The prerequisites for the workflow of the overall molecular engineering process are those of Descriptor Calculation, the first step in the process defined in [1] and [5], together with a library of fragment and core molecule structures. In addition, all necessary is available either in the data warehouse (MOLDW) or another data source like the Custom Data Repository (CDR).

As the workflows for Descriptor Calculation and Model Development have already been defined (see [5] and [6]) they are not detailed here. The steps of the workflow are the following:

1. Do Descriptor Calculation
2. Do Model Development
3. Select fragments
4. Select models
5. Execute Universal Structure Enumerator with additional parameters like the maximum number of compounds and settings for additive fragment descriptor values
6. Store ‘List of 3D Structures’ (optional 2D) and the enumeration information to CDR
7. Calculate “all” Descriptors for each structure from the ‘List of Structures’ (“all” means within OpenMolGRID all those descriptors known to Codessa)
8. Generate table with all results from the previous step
9. Query additional models from CDR and add them to the list of models prepared in step 4
10. Run Prediction (P_AP) for property prediction on List of Models, Descriptor values and parameter settings from the user for limiting the calculation
11. Select the most promising candidates from P_AP’s output (list of (structure, property value1, …, property value n, number of model used))
12. Save selected output to CDR
13. done

3.2. Dataflow Model

The overall molecular engineering process expects different input from (different) data sources and the user at different steps in the process together with the results of the previous step(s) in the workflow. The user has to provide selection criteria and parameter settings for the different query and compute steps. The description of the dataflow (Figure 2) uses the coarse-grain dataflow models given for descriptor calculation (see [5]) and model development (see [6]). For the structure generation step fragments from a structural fragment library and a list of models (generated by model development) are used. The user gives selection criteria for both and provides calculation parameters. The evaluation of the generated structures consists of adding models available from a database to the previously used list of models, calculating all descriptors for each of the generated structures, and predicting the properties for each structure and each model from the list. From the generated output the user will later select the structures with the optimal biological activity or chemical property manually to decide on which of the structures should be synthesised and tested in the laboratory.

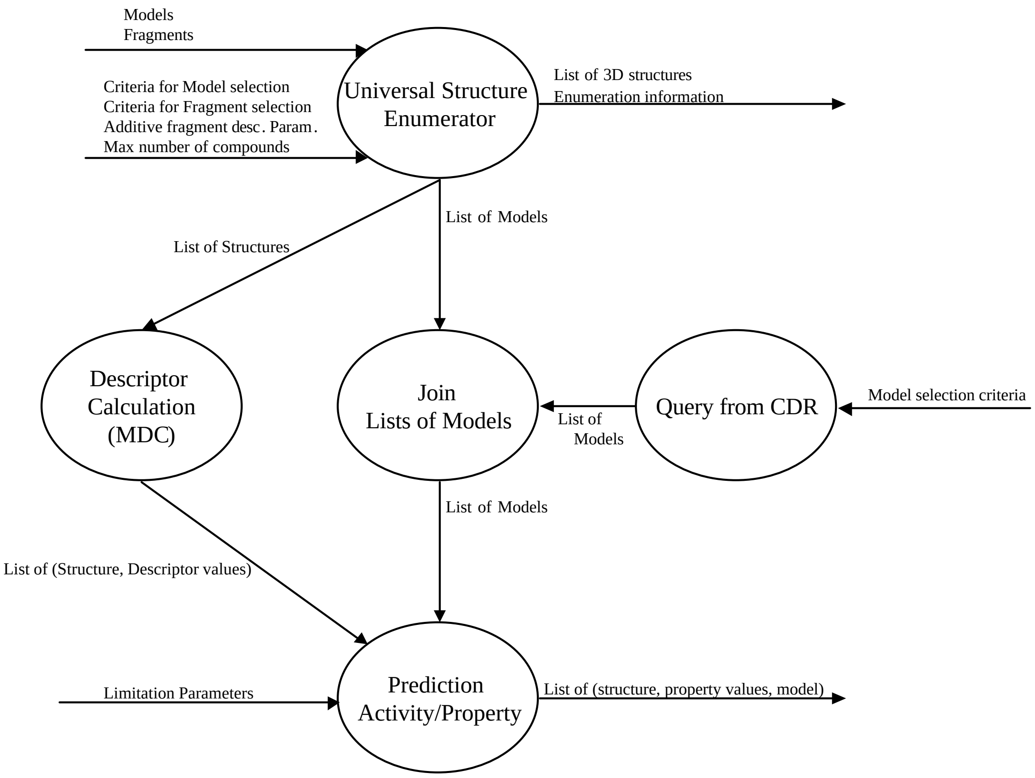
Models
Fragments
Criteria for Model selection
Criteria for Fragment selection
Additive fragment desc. Param.
Max number of compounds
Universal Structure Enumerator
List of 3D structures
Enumeration information
List of Models
List of Structures
Descriptor Calculation (MDC)
Join Lists of Models
List of Models
Query from CDR
Model selection criteria
List of Models
List of (Structure, Descriptor values)
Limitation Parameters
Prediction Activity/Property
List of (structure, property values, model)

Figure 3 shows the fine grain data flow for structure generation and screening.

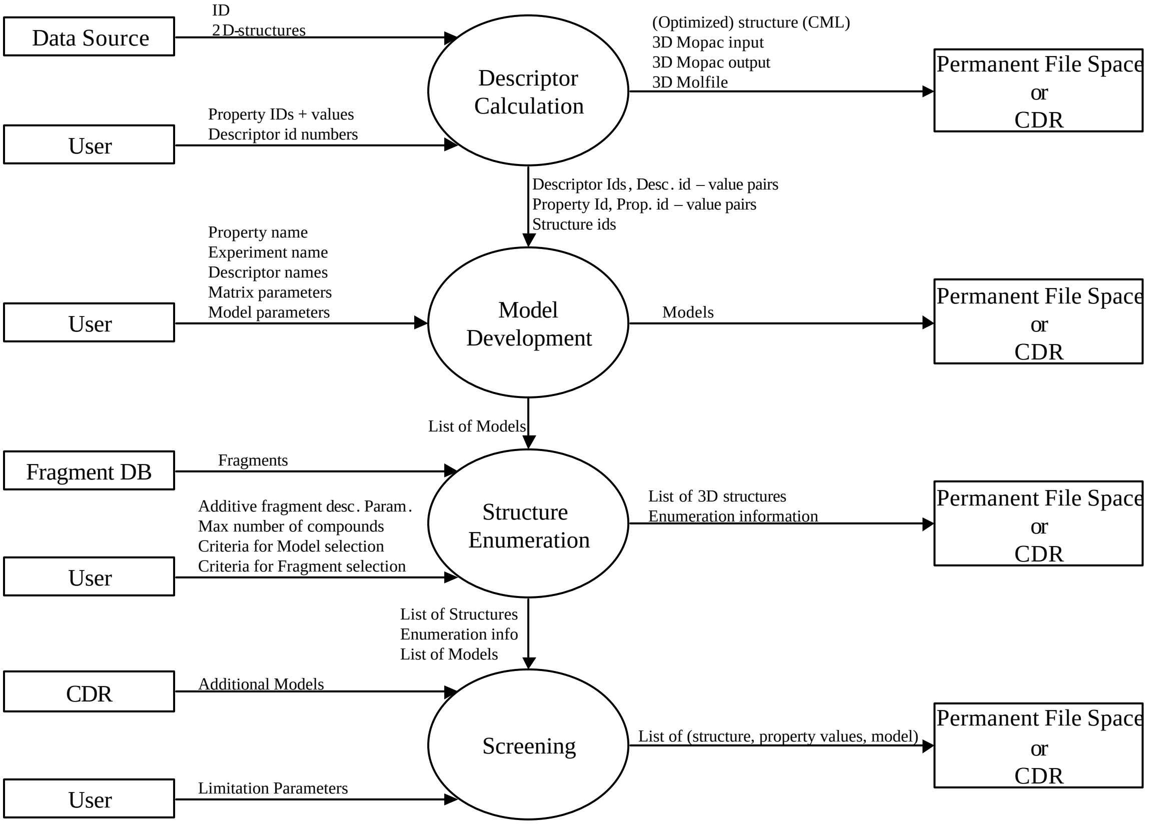

Figure 2: Coarse grain data flow of the molecular engineering process

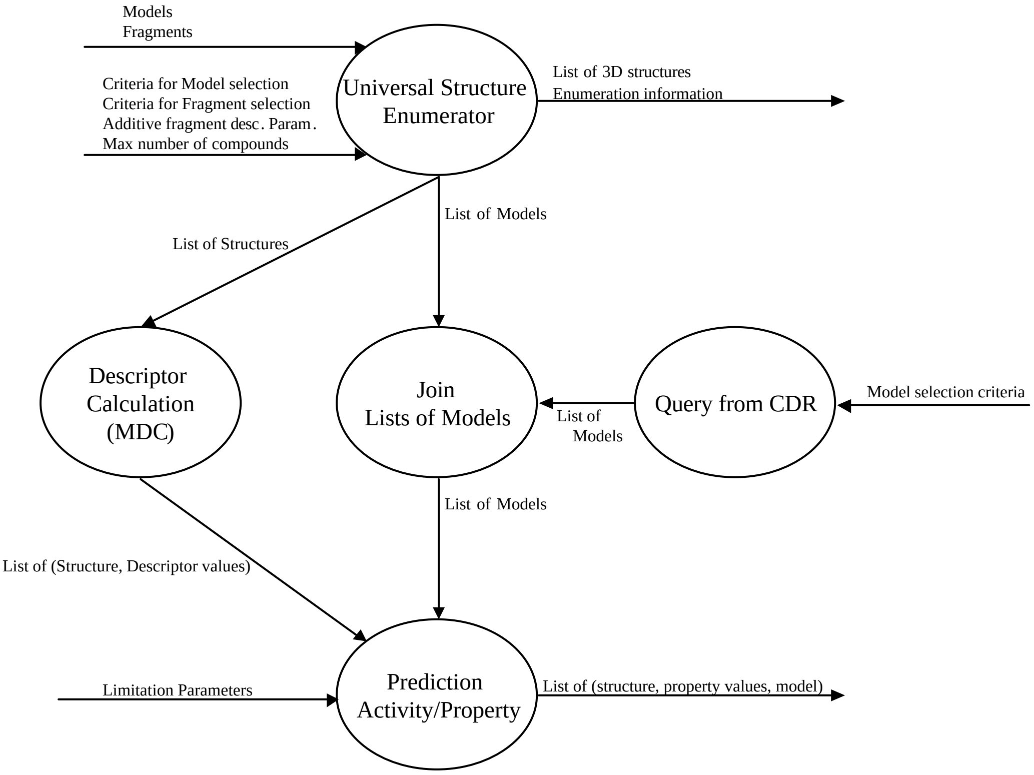
Models
Fragments
Criteria for Model selection
Criteria for Fragment selection
Additive fragment desc. Param.
Max number of compounds
Universal Structure Enumerator
List of 3D structures
Enumeration information
List of Structures
List of Models
Descriptor Calculation (MDC)
Join Lists of Models
Query from CDR
List of Models
Model selection criteria
List of Models
List of (Structure, Descriptor values)
Limitation Parameters
Prediction Activity/Property
List of (structure, property values, model)

Figure 3: Fine-grain dataflow of structure enumeration and screening

3.3. Flowchart Diagram

The flowchart for the overall process of molecular engineering given in Figure 4. The chart shows the control flow relevant to the OpenMolGRID project. The flowchart elements are defined in Figure 5. The dotted lines in the flowchart diagram separate the four major blocks of the workflow: Descriptor Calculation, Model Development, Enumeration, and Prediction. The control flow does not contain any loops as the necessary applications are all capable of dealing with a list of input records, i.e. a list of structures. Tasks like optimisation, MDC (Descriptor Calculation Module), or P_AP (Module for predicting biological activities or chemical properties) can be split to run on several target systems in parallel. The data generated in each step is forwarded to its successor step. The generated output is stored to permanent file space or the Custom Data Repository (CDR).

The control flow starts with querying a data source (the data warehouse MOLDW) for structures with certain properties. This output needs to be prepared to conform to the input format necessary for the 2D to 3D conversion. The generated 3D structures need to be optimised using semi-empirical methods. The user may want to analyse the optimised structures for usability before the structures are fed into the descriptor calculation application. The calculated descriptors per structure are passed on to the model building application The next step in the flow is the generation of structure candidates with input from the fragment database or library, models, and parameter settings. For these candidate structures the descriptors are calculated and passed on to the prediction application. The models used during enumeration are combined with additional models queried for from the CDR. The prediction runs for all structures and all models to predict the user selected properties.

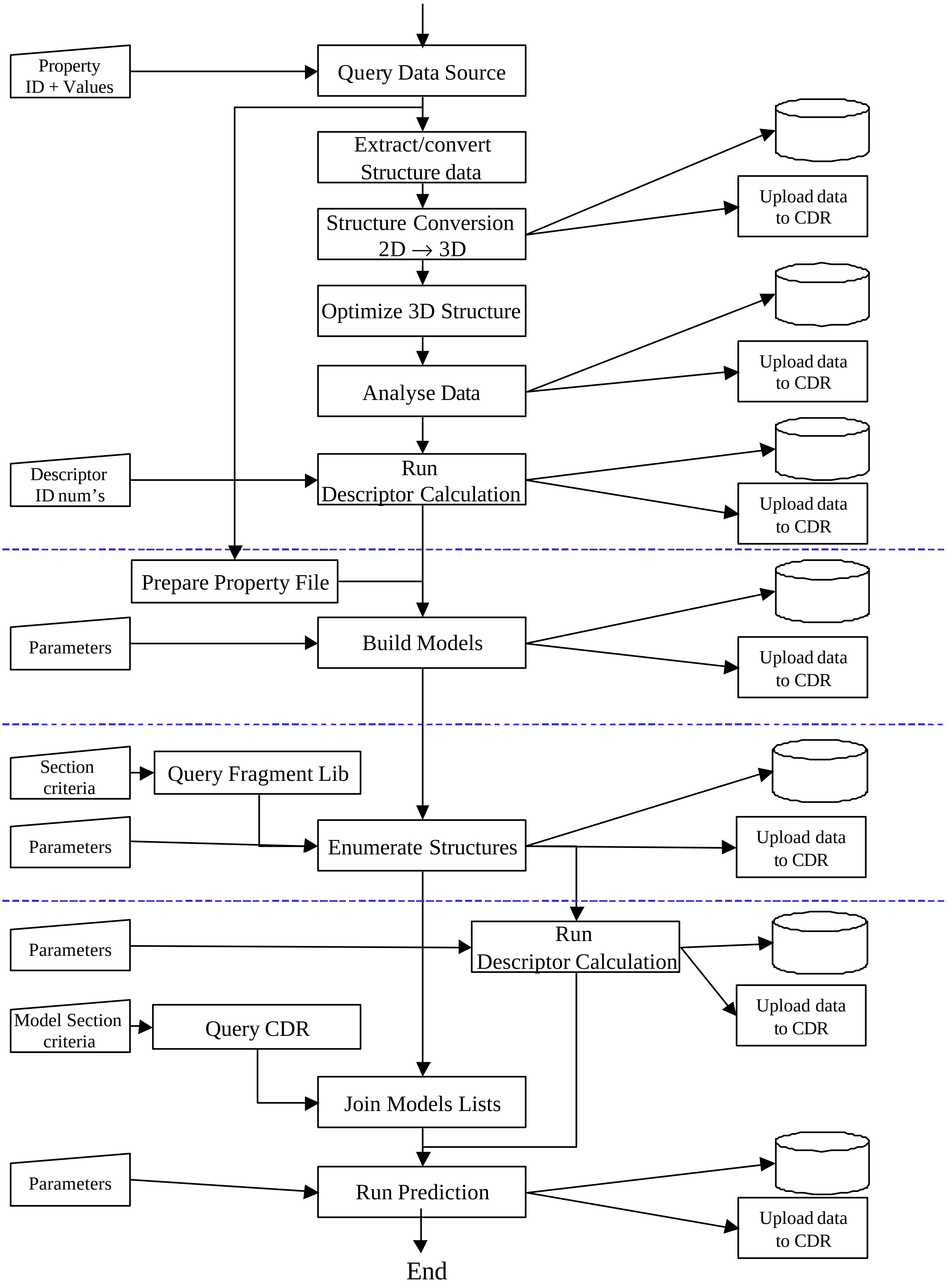

Figure 4: Flowchart for the molecular engineering process

Legend

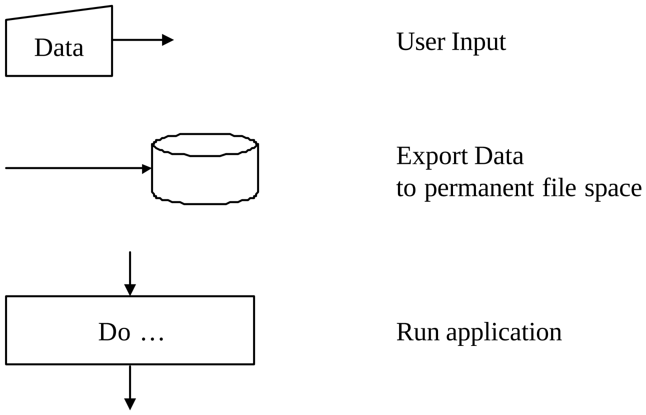

Figure 5: Description of flowchart elements

4. References

[1] Deliverable D2.1
Specification of software modules for descriptor calculation and model development and their Grid interface components

[2] Deliverable D2.4a
Description of the quantitative structure property/activity relation model: model building and application

[3] Deliverable D3.6
Description of the molecular engineering procedure

[4] Deliverable D4.2a
Specification of the Grid Interface for Classes of Applications to Support Automated Workflows

[5] Deliverable D4.2c
Specification of the Workflow for Descriptor Calculation

[6] Deliverable D4.3
Specification of the workflow for QSPR/QSAR model development

5. Terminology / Glossary

CDR	Custom Data Repository
CGX	ComGenex
FZJ	Forschungszentrum Jülich
MOLDW	OpenMolGRID Data Warehouse
Negri	Istituto di Ricerche Farmacologiche Mario Negri
$\mathbf{P_AP}$	Software Module for Prediction of Biological Activity and Chemical Property
UNICORE	Uniform Interface to Computer Resources
UT	University of Tartu
UU	University of Ulster
WP	Work Package
XML	Extensible Markup Language

Annex 1: XML Specification of the Molecular Engineering Workflow

In the following the workflow is given in XML workflow format as defined in [4]. This specification can directly be used as input for the MetaPlugin which is the UNICORE Client Plugin for OpenMolGRID workflow handling.

```
<?xml version="1.0" ?>
<!DOCTYPE workflow [
<!ELEMENT workflow (task*, group*, dependency*)>
    <!ELEMENT task (option*)>
      <!ELEMENT option EMPTY>
        <!ATTLIST option
             name     CDATA #REQUIRED
             value    CDATA #REQUIRED
        >
      <!ATTLIST task
           name        CDATA #REQUIRED
           identifier  CDATA #REQUIRED
           id          CDATA #REQUIRED
           export      (true | false) #REQUIRED
           split       (true | false) #REQUIRED
           splitterTask CDATA
           joinerTask   CDATA
      >
    <!ELEMENT group (option*, task*, group*, dependency*)>
      <!ATTLIST group
           type        (subjob | repeat | doN | if | then | else) #REQUIRED
           identifier  CDATA #REQUIRED
           id          CDATA #REQUIRED
           split       (true | false) #REQUIRED
      >
    <!ELEMENT dependency EMPTY>
      <!ATTLIST dependency
           pred        CDATA #REQUIRED
           succ        CDATA #REQUIRED
      >
]>

<workflow>
<!-- workflow for molecular engineering combining descriptor calculation, model
development, structure enumeration, and prediction -->

<group type="subjob" identifier="Query Database" id="1">
       <!-- wrapper group to allow easy datasource selection -->
  <task name="DataBaseRequest" identifier="Query Database" id="110"
       export="false" split="false">
  </task>
</group>

<task name="DataBaseRequestToSLF" identifier="Structure file preparation" id="2"
    export="false" split="false">
</task>

<task name="2Dto3Dconversion" identifier="Convert 2D to 3D" id="3"
    export="false" split="false">
</task>
    <task name="DataBaseSave" identifier="Save_to_CDR" id="4" export="false"
split="false">
    </task>

<task name="SemiempiricalCalculation" identifier="Structure optimization" id="5"
    export="false"
    split="true" splitterTask="SplitStructureList"
    joinerTask="JoinStructureLists">
    <option name="keywords" value="AM1 EF PRECISE"/>  <!-- define mopac params
here -->
</task>

<task name="Analyse3DStructure" identifier="Check Structures" id="6"
    export="false" split="false">
</task>

<task name="DataBaseSave" identifier="Save_to_CDR" id="7" export="false"
split="false">
    </task>

<task name="DescriptorCalculation" identifier="Codessa descriptor calculation"
id="8"
    export="false" split="true">
</task>

    <task name="DataBaseSave" identifier="Save_to_CDR" id="9" export="false"
split="false">
    </task>

<task name="DataBaseRequestToPLF" identifier="Property file preparation" id="10"
    export="false" split="false">
</task>

<task name="ModelBuilding" identifier="Model building" id="11"
    export="false" split="false">
</task>

<task name="DataBaseSave" identifier="Save_to_CDR" id="12" export="false"
split="false">
    </task>

<task name="QueryFragmentLib" identifier="Select Fragments" id="13"
export="false" split="false">
    </task>

<task name="UniversalStructureEnumerator" identifier="Enumeration" id="14"
export="false" split="yes">
    </task>

<task name="DataBaseSave" identifier="Save_to_CDR" id="15" export="false"
split="false">
    </task>

<task name="DescriptorCalculation" identifier="Codessa descriptor calculation"
id="16"
    export="false" split="true">
</task>

<task name="DataBaseSave" identifier="Save_to_CDR" id="17" export="false"
split="false">
    </task>

<group type="subjob" identifier="Query Database" id="18">
       <!-- wrapper group to allow easy datasource selection -->
  <task name="DataBaseRequest" identifier="Query Database" id="180"
       export="false" split="false">
  </task>
</group>

<task name="JoinModelLists" identifier="Prepare list of Models" id="19"
export="false" split="false">
    </task>
```

```
<task name="PAP" identifier="Prediction" id="20" export="false" split="false">
    </task>


<task name="DataBaseSave" identifier="Save_to_CDR" id="21" export="false"
split="false">
    </task>

<dependency pred="1" succ="2"/><!-- db request to structure extract-->
<dependency pred="1" succ="10"/><!-- db request to property extract-->
<dependency pred="2" succ="3"/><!-- struct extract to 2d to 3d -->
<dependency pred="3" succ="4"/><!–store data-->
<dependency pred="3" succ="5"/><!-- 2d to 3d to semi-empirical-->
<dependency pred="5" succ="6"/><!-- semiempirical to analysis-->
<dependency pred="6" succ="7"/><!-- analysis to descriptor calc-->
<dependency pred="6" succ="8"/>
<dependency pred="8" succ="9"/><!–store data-->
<dependency pred="8" succ="11"/>
<dependency pred="10" succ="11"/>
<dependency pred="11" succ="12"/><!–store data-->
<dependency pred="11" succ="14"/>
<dependency pred="13" succ="14"/>
<dependency pred="14" succ="15"/><!–store data-->
<dependency pred="14" succ="16"/>
<dependency pred="14" succ="19"/>
<dependency pred="16" succ="17"/><!–store data-->
<dependency pred="16" succ="20"/>
<dependency pred="18" succ="19"/>
<dependency pred="19" succ="20"/>
<dependency pred="20" succ="21"/><!–store data-->

</workflow>
```