INFORMATION SOCIETY TECHNOLOGIES
(IST)
PROGRAMME

OpenMolGRID

SPECIFICATION OF THE WORKFLOW FOR QSPR/QSAR MODEL DEVELOPMENT

Contract Reference:	**IST-2001-37238**
Document identifier:	**OpenMolGRID-4-D4.3-0111-0-3-WorkflowMD**
Date:	**31/01/2004**
Work package:	**WP 4: Grid Integration**
Partner:	**UT, UU, Negri, FZJ, CGX**
Lead Partner:	**FZJ**
Document status:	**APPROVED**
Classification:	**PUBLIC**
Deliverable identifier:	**D4.3**

Abstract: Specification of the workflow to be provided for Model Development

Delivery Slip

	Name	Partner	Date
From	Mathilde Romberg	FZJ	07/01/2004
Verified by	Mathilde Romberg	FZJ	07/01/2004
Approved by	G.H.F.Diercksen (TC)	OMC	25/01/2004
	R.Ferenczi (QE)	CGX	23/01/2004

Document Log

Issue	Date	Comment	Author
0-0	03/12/2003	initial version	M.Romberg, B.Schuller
0-1	06/01/2004	update	M.Romberg
0-2	07/01/2004		M.Romberg
0-3	31/01/2004	approved version	M.Romberg

Document Change Record

Issue	Item	Reason for Change
0-1	2.3, annex	incomplete information
0-2	all; MD process refined	input from UT
0-3	2	input from CGX, TC

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 n/Deliverables/ OpenMolGRID-4-D4.3-0111-0-3-WorkflowMD

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

Molecular descriptors are used to characterize molecular structures and thereby provide the basis for a measure for molecular similarity. QSPR/QSPR model building (see [1]) uses the descriptor values of compounds to generate predictive models for the estimation of chemical property or biological activity values. In this document the workflow for model building based on available descriptor values is modelled within the Grid infrastructure UNICORE ([2]).

Workpackage 2, Molecular Descriptor Generation and QSPR Model Building on the Grid, will adapt multiple existing software packages for the QSPR/QSAR analysis to be executed in the Grid environment. Client plugins and adapter components will be developed. Here we are specifying the workflow for QSAR/QSPR model building which will be used as input to the MetaPlugin specified in [3].

1.2. Overview

OpenMolGRID is mainly interested in statistical models based on descriptor values related to the 3D structure of a compound. Therefore the model building process uses the data generated during descriptor calculation (see [4]). The workflow for QSPR/QSAR model building comprises two major steps: the generation of the condensed compound - descriptor value - matrix and the model building itself. The workflow with support from the MetaPlugin ([3]) will guide the user through the process and hide all tasks which can be prepared and done automatically (i.e. data conversion, data transfer) away from the user. The knowledge about prerequisites of applications come from the metadata attached to the corresponding UNICORE application resource.

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 is attached.

1.3. Document Structure

The document contains in Section 2 the different views of the workflow, Section 3 and Section 4 hold references and glossary, and Appendix 1 gives the XML representation of the workflow.

2. Workflow Specification

The goal of the processes under consideration is to find compounds with certain property and/or activity values. From experience it is known that molecules with similar descriptor values have similar chemical property and biological activity values. The prediction model established by the workflow under consideration in this deliverable correlates descriptor values to properties and activities. The major input for it are the descriptor values calculated in the previous step ([4]).

Software packages (SW packages) exist to do model building. The SW packages relevant to OpenMolGRID (e.g. Codessa mda) are integrated into UNICORE in WP2. In WP4 the workflow for model development is integrated. The UNICORE Client plugin on control structure level designed in [3] will be used here and fed with a workflow, the one for model building. The workflow has been compiled from the WP2 use cases ([5]) by the project partners UT, CGX, UU, and FZJ in a face-to-face meeting in April 2003.

2.1. Workflow Description

The prerequisite is that all necessary data (structure information, descriptor id -value pairs, descriptor names, descriptor ids, property names, property ids, and property id – value pairs) is available in the data warehouse (MOLDW) or the custom data repository (CDR) or another data source. The verbal description of the workflow contains the steps to be processed together with examples of software packages and corresponding data formats.

The following steps are processed by the user to do model development:

1. Query property id for a property name
2. Query descriptor ids for a set of descriptor names
3. Query structure ids of structures with the relevant property id
4. Query descriptor id – value pairs for the set of descriptor ids per structure id
5. Query property id – value pair for the relevant property id and experiment per structure id
6. Get parameters for matrix generation from user input and generate input matrix for model building (output is a is a delimited text file compatible with STATISTICA)
7. Get parameters for model building from user input and build models (e.g. run Codessa on matrix for model building; generates *n* files containing one model each in PMML format)
8. Store output (list of files with one model each) to permanent file space and/or CDR
9. Done

2.2. Dataflow Model

The workflow for model building is mainly determined by the data to be exchanged between the applications. Therefore, the dataflow for the model building process is specified in two diagrams, first the high-level dataflow showing the data sources, the relevant input data, the produced output data, and its destination. The second diagram shows the internals of the model building process. Each circle contains an application which wraps a SW package, the arrows show the direction of the dataflow and which data goes from one step to the other. Data coming from data sources goes in without specifying the sources but it matches exactly what is specified in the high-level dataflow diagram. The same holds for output data.

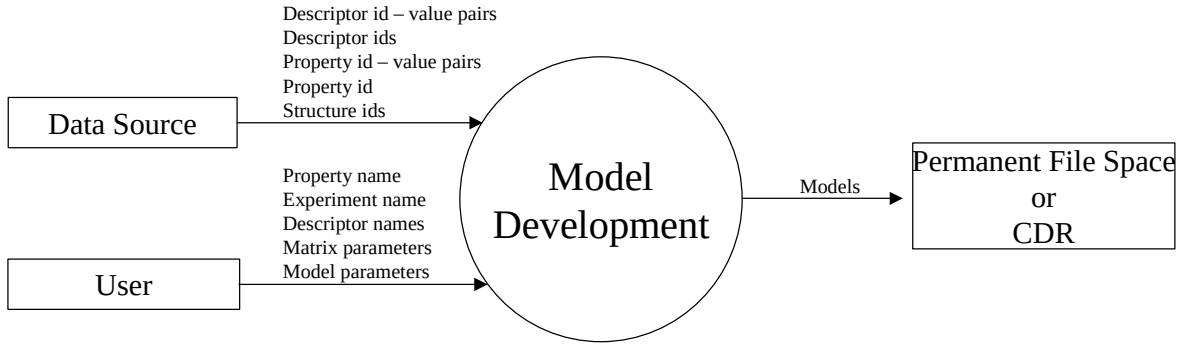

Figure 1: Top-level dataflow model for Model Development

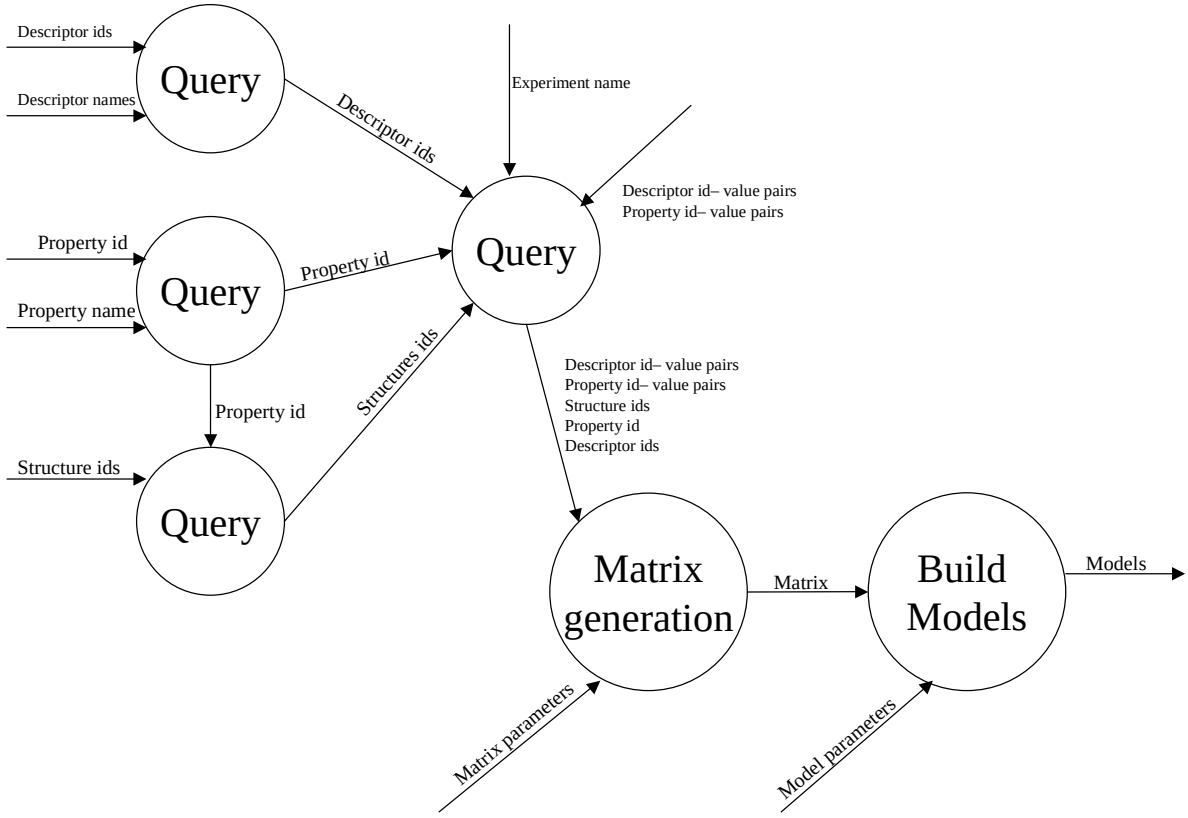

Figure 2: Detailed dataflow model for Model Development

2.3. Flowchart Diagram

The flowchart for the process of model development given in Figure 3 is quite simple as it does not contain any loops nor manual user intervention nor tasks which can be split to run on several target systems in parallel. The user provides the input for querying the data source and selects descriptor names from a list available from a data source. The data generated in each step is forwarded to its successor step. The generated models are stored to permanent file space.

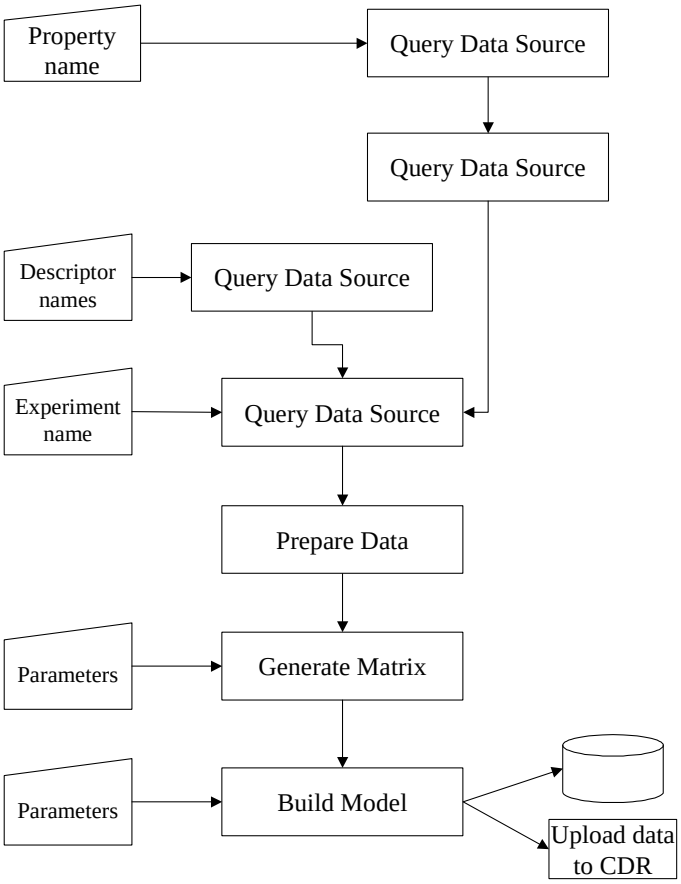

Figure 3: Model Development Flowchart

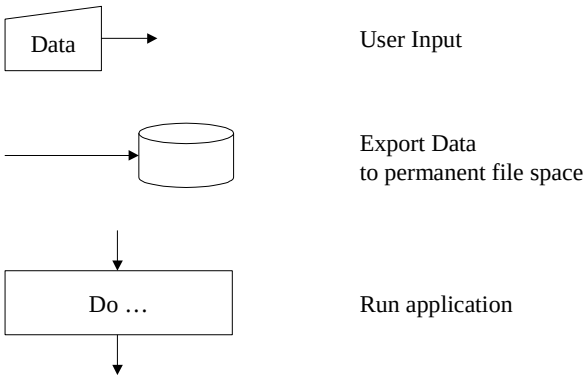

Figure 4: Legend for Flowchart Diagram

3. References

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

[2] Deliverable D4.5a,
Description of the OpenMolGRID Grid architecture, security architecture, and infrastructure and the deployment of the project's testbed

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

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

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

4. Terminology / Glossary

AJO	Abstract Job Object
CAS	Chemical Abstracts Service
DBAT	Database Access Tool
FZJ	Forschungszentrum Jülich
GUI	Graphical User Interface
JRE	Java Runtime Environment
MOLDW	OpenMolGRID Data Warehouse
NJS	Network Job Supervisor
NTP	National Toxicity Program
SQL	Structured Query Language
UNICORE	Uniform Interface to Computer Resources
UU	University of Ulster
WP	Work Package
XML	Extensible Markup Language

Annex 1: XML Specification for MD Workflow

The following XML document contains the Document Type Definition together with the representation of the workflow for QSPR/QSAR model development. The task names chosen in the workflow are preliminary as they correspond to the task identifier the applications provide as part of their metadata. The list of task names agreed on in the project will be compiled later.

```
<?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
      >
    <!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>
  <!-- specification of Model Development workflow -->

  <task name="DataBaseRequest" identifier="Get_PropId" id="1" export="false"
split="false">
  </task>
  <task name="DataBaseRequest" identifier="Get_DescIDs" id="2" export="false"
split="false">
  </task>
  <task name="DataBaseRequest" identifier="Get_StructureIDs" id="3"
export="false" split="false">
  </task>
  <task name="DataBaseRequest" identifier="Get_Values" id="4" export="false"
split="false">
  </task>
  <task name="ExtractDataFromDataBaseRequest" identifier="DataPreparation"
id="5" export="false" split="false">
  </task>
  <dependency pred="1" succ="3" />
  <dependency pred="2" succ="4" />
  <dependency pred="3" succ="4" />
  <dependency pred="4" succ="5" />
  <task name="MatrixGeneration" identifier="Matrix" id="6" export="false"
split="false">
  </task>
```

```
  <task name="ModelDevelopment" identifier="Model" id="7" export="true"
split="false">
  </task>
  <dependency pred="5" succ="6" />
  <dependency pred="6" succ="7" />
</workflow>
```