

OpenMolGRID: Using Automated Workflows in GRID Computing Environment

Sulev Sild¹, Uko Maran¹,
Mathilde Romberg², Bernd Schuller²
Emilio Benfenati³

¹ Department of Chemistry, University of Tartu, Tartu 51014, Estonia
{sulev.sild, uko.maran}@ut.ee

² Forschungszentrum Jülich GmbH, ZAM, D-52425 Jülich, Germany
{m.romberg, b.schuller}@fz-juelich.de

³ Istituto di Ricerche Farmacologiche "Mario Negri" Via Eritrea 62, 20157 Milano, Italy
benfenati@marionegri.it

Abstract. Quantitative Structure Activity/Property Relationship (QSAR/QSPR) model development is a complex and time-consuming procedure involving data gathering and preparation. It plays an important role in the drug discovery pipeline, which still is mostly done manually. The current paper describes the automated workflow support of the OpenMolGRID system and provides a case study for the automation of the QSPR model development process in the Grid.

1 Introduction

The typical process for solving complex scientific problems involves the execution of time-consuming tasks that have to be carried out in a specific order. Often these interdependent tasks are carried out by independent applications that may require the manual processing of intermediate results by end-users in the middle of the process. While Grid computing provides a powerful infrastructure for making distributed computational resources available to compute intensive tasks, the automated workflows provide new ways to combine otherwise independent programs (or services) in the Grid environment to create a new form of applications designed specifically for the problem at hand. This article describes the automated workflow support of the OpenMolGRID system and provides a case study for the automation of the QSPR model development process.

1.1 Open Computing Grid for Molecular Science and Engineering

Open Computing Grid for Molecular Science and Engineering (OpenMolGRID) [1] is a project focused on the development of Grid enabled molecular design and engineering applications. *In silico* testing has become a crucial part in the molecular de-

sign process of new drugs, pesticides, biopolymers, and biomaterials. In a typical design process hundred thousands or even millions of candidate molecules are generated and their viability has to be tested. Economically it is not feasible to carry out an experimental testing on all possible candidates. Therefore, computational screening methods provide a cheap and cost effective alternative to reduce the number of candidates to a more manageable size. Over the years quantitative structure activity/property relationship (QSAR/QSPR) methods have been proved to be a reliable for the prediction of various physical, chemical and biological activities [2, 3].

The QSAR/QSPR methodology relies on a basic assumption that biological activity or physical property is a function of the molecular structure [4]. The molecular structure is characterized by theoretical parameters or so called molecular descriptors. Various statistical [5, 6] and variable selection [7] methods are then used to find quantitative relationships between experimentally available biological activity data and relevant molecular descriptors. The development of QSAR/QSPR models is rather complicated in practice, since various data pre-processing steps are required to prepare a proper training set before the model development can be started. The most common pre-processing tasks include the collection of experimental data, the generation of 3D coordinates for input structures, quantum chemical calculations, and molecular descriptor calculation. All these steps must be repeated in a proper sequence for each molecule in the training set. Traditionally, this kind of workflow involves a lot of manual labor and user interaction that is not practical when huge data sets are processed. The consistency of the data set is very important and the manual process may introduce unnecessary errors due to human factors. Computations in the data processing steps are time consuming, especially when quantum chemical calculations are involved.

The molecular design process can be significantly improved when Grid resources are exploited for the development and application of QSAR/QSPR models. Improvements are possible both for the reduced time necessary for the task of building a model (with all the above listed steps) and better quality resulting from the automation, which reduces mistakes and the variability of results. The OpenMolGRID project addresses the above-described problems by providing a Grid enabled infrastructure to automate these complex workflows and to speed up the process by distributing data parallel tasks over available computational resources.

2 Automated Workflows in Grid Environment

The specification and execution of complex processes like the process of molecular design and engineering using Grid resources is still an open field in Grid research and development. Solutions exist mostly for business processes. Languages to describe business processes are for example BPEL4WS (Business Process Execution Language for Web Services, see [8]) and WPD (Workflow Process Definition Language, see [9]). The modeling of complex workflows in the scientific arena is mostly done manually using the tools some Grid middleware offers. The key point is the description of software resources available on Grid computing resources. These descriptions can be used for automated application identification and inclusion in multi-

step workflows. The following sub-sections will describe the solution for automated workflow specification and processing developed within the OpenMolGRID project.

2.1 Workflow Specification

The primary question to be answered for identifying the necessary elements for a closed definition of a workflow is how applications and their interfaces are described in the environment at hand. In general workflows are built of tasks or processes as key elements. These elements are related through sequential temporal dependencies correlated to data flow between them or they are independent. OpenMolGRID uses the UNICORE Grid middleware [10] which offers workflow specification within a graphical user interface where tasks and sub-jobs are graphically linked to reflect dependencies and the necessary data flow is given through explicit transfer tasks. In addition workflow elements like loops, if-the-else, and hold are available to build up complex jobs. Applications or tasks within a job or sub-job are available on the client side as (application specific) plugins, which correspond to defined application resources on the server side. These resources are described by metadata that, among others, define the interface and I/O format information to clients.

Existing workflow description languages do not match the UNICORE model with respect to software resources. As these play the most important role within the automatic job generation a workflow specification language has been developed which allows a high level definition of various scientific processes containing sufficient information for the automatic generation of complete UNICORE jobs. This includes the tasks and their dependencies but also necessary resources for the steps. XML has been selected as a specification language for the workflow. A core element in a workflow is *task*, which has

- a *name* giving the identifier of a task fulfilled by an application resource and supported by a Client Plugin,
- an *identifier* giving the name for the UNICORE task in the job tree,
- an *id* giving the unique numerical identification within the workflow,
- an *export* flag specifying whether result files are to be exported to the user's workstation,
- a *split* flag specifying whether the task is data parallel and can be distributed onto several execution systems,
- a *splitterTask* giving the name of an application which is capable of splitting the input data for this task into n chunks,
- a *joinerTask* giving the name of an application which is capable of joining the n result files into one file, and
- *options* to feed the application with parameter settings.

For a *task* a set of simple resources can be specified requesting *runTime*, number of *nodes*, number of *processorsPerNode*, and *memoryPerNode*. For a *group* element of the workflow, which corresponds to a UNICORE sub-job the target system for the execution of all tasks within the group can be specified by *usite* and *vsite*. The workflow specification details are given in Appendix A.

Currently, there is no additional tool to generate the XML workflow; one has to use a standard text editor. With new UNICORE Client developments this will change

2.2 Workflow Processing

A workflow specified as described above serves as input to the MetaPlugin, a special Plugin to the UNICORE Client. The MetaPlugin parses the XML workflow, creates a UNICORE job from it, and assigns target systems and resources to it. These tasks include a lot of sophisticated actions:

- Sub-jobs have to be introduced into the job wherever necessary, for example when requested applications are not available on the same target system;
- Transfer tasks have to be introduced into the job to ship data from one target system to another, which is target of a sub-job;
- Data conversion tasks have to be added between two tasks where the output format (specified in XML according to the application metadata) of one task does not match the input format of the successor task;
- Splitter and transfer tasks have to be added to the workflow as predecessor tasks of a splittable task for input data preparation;
- Sub-jobs have to be created around splittable tasks for each selected target system, and a transfer task to transfer the output data back to the superordinate sub-job;
- Joiner tasks have to be added to join the output data of split tasks;
- The directed acyclic graph of dependencies between all tasks (the explicit ones from the workflow specification and the automatically generated ones) has to be set up.

The MetaPlugin uses the resource information provided by the target system (*vsite*), the metadata of the applications, and information about the Plugins available in the Client. A so called resource information provider component has been developed to support the MetaPlugin in resource selection: It says which Client Plugin serves the task, which target system offers the application, and which are the I/O formats. Currently the MetaPlugin does resource selection at a very basic level but a more sophisticated resource broker component could easily be added.

The main advantage of this mechanism is that a user who wants to do model building can name the coarse-grained tasks and their dependencies in an XML workflow thereby avoiding the tedious job of the step-by-step preparation of the UNICORE job. The latter would afford detailed knowledge about for instance I/O formats for correct job preparation and the manual splitting and distribution of tasks onto appropriate target systems. Doing this automatically gives a lot of flexibility to the system to adapt to the actual Grid layout and resource availability and it helps avoiding human errors.

3 Case Study: Prediction of Solubility

The solubility is one of the most significant properties of chemicals and therefore important in various areas of human activity. Solubility in water is fundamental to environmental issues such as pollution, erosion, and mass transfer. Solubility in organic solvents forms much of the basis of the chemical industry. Solubility determines shelf life and cross contamination. Toxicity is critically dependent on solubility. Solubility is also linked to bioavailability and thus to the effectiveness of pharmaceuticals. Solubility is one of the most important parameters of the ADME/Tox (absorption, distribution, metabolism, elimination and toxicity) profile that is used to test the drug ability (drug-likeness) of potential new drugs [11].

Therefore the computational prediction of solubility has been of huge interest, and the methods range from statistical and quantum mechanics to QSPR approaches [12]. The latter method is implemented in the OpenMolGRID system, an environment for solving the large-scale drug design and molecular design problems.

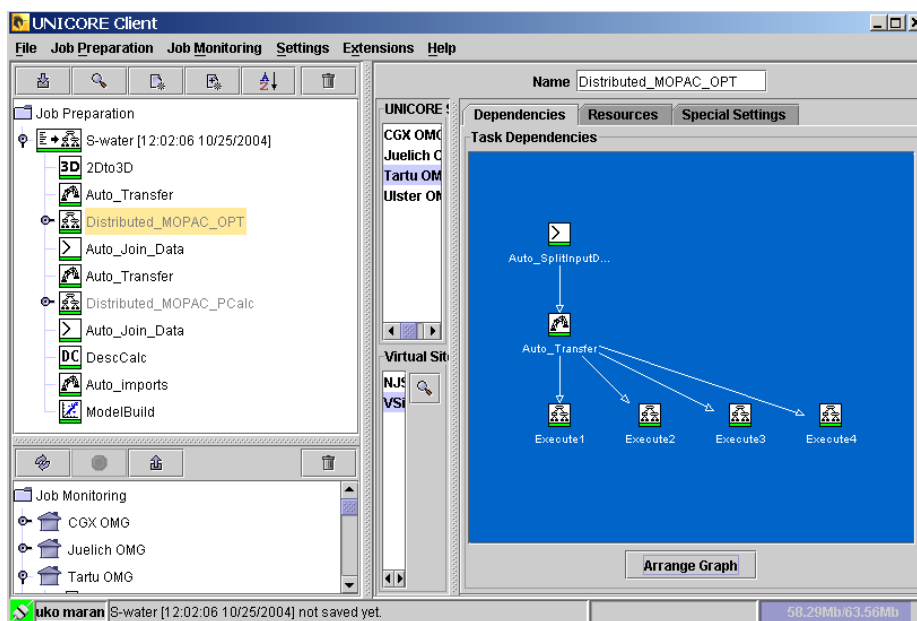


Fig. 1. Graphical representation of the OpenMolGRID workflow

3.1 Description of the Solubility Data and Distributed Tasks

The example of using workflows in the Grid environment is given with the prediction of solubility in water. The example set of 178 data points consists of a large range of organic chemicals and is described in detail elsewhere [12].

The development of a QSPR model for the prediction of water solubility involves 4 distributed tasks in our Grid environment: (i) the 2D to 3D conversion of molecular structures; (ii) semi-empirical quantum chemical calculations of 3D structures; (iii) calculation of molecular descriptors for each 3D structure; and (iv) building up QSPR models. All those tasks are mapped onto geographically distributed resources and do need different amounts of computational resources, with the first and second step being the most demanding.

The XML workflow used for the prediction of the solubility is given in Appendix B. As one can see the semi-empirical tasks are automatically split between available computational resources. Also different options can be set for the tasks. In the current example a predefined set of keywords for the semi-empirical calculations is specified. It is not necessary to set them by hand when the workflow is reused. The graphical representation of the full workflow is given in the UNICORE Client Job Preparation area and the splitting of the semi-empirical task can be seen in the Task Dependencies area in Figure 1.

3.3 The QSPR Model for the Prediction of Water Solubility

The above described model development workflow has been carried out with the OpenMolGRID system and it produced a multi-linear QSPR equation (Table 1) with five descriptors for the prediction of the water solubility.

Table 1. The developed 5-descriptor QSPR model

Descriptors	Coefficient	<i>t</i> -test
Intercept	-0.81906	-5.48578
count of H-acceptor sites (MOPAC PC)	1.95510	22.60995
LUMO+1 energy	-0.27144	-7.77956
Min partial charge (Zefirov PC)	-13.80021	-13.18722
Number of rings	0.83094	8.03049
HA dep. HDSA-2/TMSA (MOPAC PC)	27.02062	7.32149

The squared correlation coefficient, R^2 , of this model equals to 0.94, its squared cross-validated correlation coefficient R^2_{CV} equals to 0.93, its F-value equals to 517.27, and its standard error of estimate, s , equals to 0.56. The R^2 and R^2_{CV} values are close to each other showing good predictive potential of the model. The analysis of the *t*-test values reveals that for the solubility the most important characteristics of the molecular structure are hydrogen bonding and minimum partial charge. Other characteristics of the structure are less important but influential like the number of aromatic and aliphatic rings in molecules. The plot of experimental versus predicted solubility values is given in Figure 2 together with the Job Monitoring area showing the successfully finished workflow.

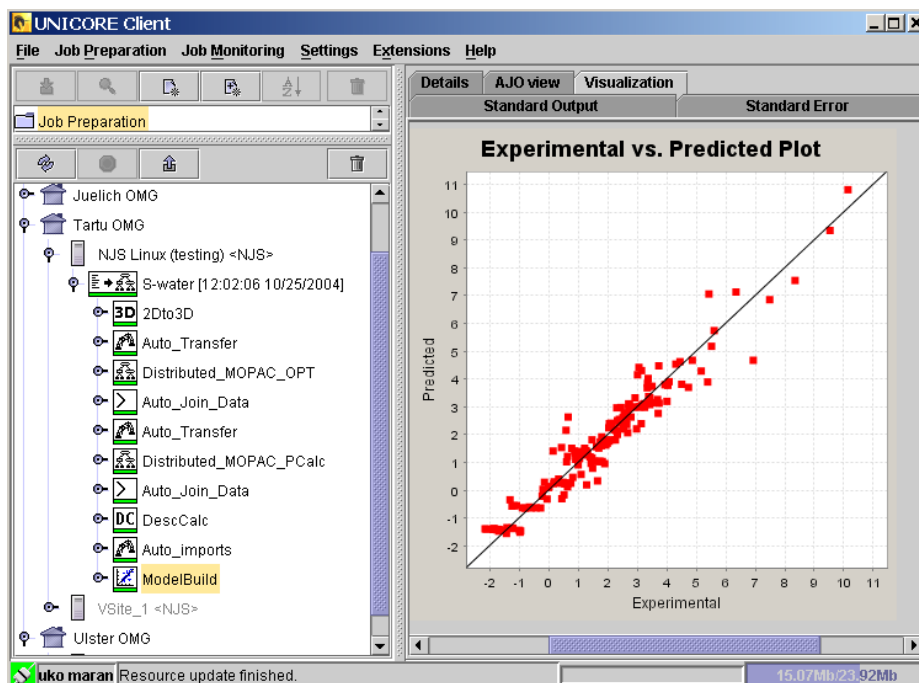


Fig. 2. Job Monitoring area with the finished workflow for Model Development and respective plot for 5-descriptor QSPR.

4 Conclusions

The automated workflows in the Grid environment offer many attractive features to the end-users in many application domains, not limited to the molecular design. They reduce the time used for manual repetitive operations and allow convenient and transparent use of distributed resources. The automated workflows are user friendly and reduce the probability for human errors. In the above described model development process, the system located appropriate resources to carry out all the required tasks and automatically handled time consuming data conversion and transfer operations that normally involve manual processing. Workflows follow a unique defined procedure (once fixed) and thus it eliminates the problem of variability, related to the subjective choice of input parameters done by different users. For the above reasons, the results obtained with predefined workflows are easier to reproduce, a characteristic, which is very valuable in general and in particular for regulatory purposes (e.g. the assessment of toxicity by regulatory bodies).

5 Acknowledgements

Financial support is greatly acknowledged from the EU 5-th framework Information Society Technologies program (grant no. IST-2001-37238).

References

1. <http://www.openmolgird.org>
2. Katritzky, A. R., Maran, U., Lobanov, V. S., Karelson, M.: Structurally Diverse QSPR Correlations of Technologically Relevant Physical Properties. *J. Chem. Inf. Comput. Sci.* 40 (2000) 1-18
3. Katritzky, A. R., Fara, D. C., Petrukhin, R., Tatham, D. B., Maran, U., Lomaka, A., Karelson, M.: The Present Utility and Future Potential for Medicinal Chemistry of QSAR/QSPR with Whole Molecule Descriptors. *Curr. Top. Med. Chem.* 2 (2002) 1333-1356
4. Karelson, M.: *Molecular Descriptors in QSAR/QSPR*. John Wiley & Sons, New York (2000).
5. Kowalski, B. R. (ed.): *Chemometrics: Mathematics and Statistics in Chemistry*. Nato Science Series: C, Vol. 138. Kluwer Academic Publishers (1984)
6. Leardi R.: *Nature-Inspired Methods in Chemometrics: Genetic Algorithms and Artificial Neural Networks*. Elsevier Science (2003)
7. Maran, U., Sild, S.: QSAR Modeling of Mutagenicity on Non-congeneric Sets of Organic Compounds. In: Dubitzky, W., Azuaje, F. (eds.): *Artificial Intelligence Methods and Tools for Systems Biology*, Kluwer Academic Publishers, Boston Dordrecht London (2004) 19-36
8. Business Process Execution Language for Web Services. Version 1.0. 31-July-2002. (<http://www-106.ibm.com/developerworks/library/ws-bpel1/>) By Francisco Curbera (IBM), Yaron Goland (BEA Systems), Johannes Klein (Microsoft), Frank Leymann (IBM), Dieter Roller (IBM), Satish Thatte (Microsoft - Editor), and Sanjiva Weerawarana (IBM). Copyright 2001-2002 BEA Systems, International Business Machines Corporation, Microsoft Corporation, Inc.
9. zur Muehlen, Michael; Becker, Jörg: WPD L – State-of-the-Art and Directions of a Meta-Language for Workflow Processes. In: Bading, L. et al. (Ed.): *Proceedings of the 1st Know-Tech Forum, September 17th-19th 1999, Potsdam 1999*
10. <http://unicore.sourceforge.net/paper.html>
11. Butina, D., Segall, M. D., Frankcombe, K.: Predicting ADME properties *in silico*: methods and models. *Drug Discovery Today* 7 (2002) S83-S88
12. Katritzky, A. R., Oliferenko, A. A., Oliferenko, P. V., Petrukhin, P., Tatham, D. B., Maran, U., Lomaka, A., Acree, W. E. Jr.: A General Treatment of Solubility. Part 1. The QSPR Correlation of Solvation Free Energies of Single Solutes in Series Solvents. *J. Chem. Inf. Comput. Sci.* 43 (2003) 1794-1805

Appendix A

XML Data Type Defintion for the specification of workflows:

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

Appendix B

The XML workflow used for the prediction of solubility

```
<?xml version="1.0"?>
<!-- Model development for Solubility in Water -->
<workflow
xmlns="http://www.openmolgrid.org/namespaces/2004/WorkflowDescription"
xmlns:rd="http://www.openmolgrid.org/namespaces/2004/SimpleResources">
<task name="2Dto3Dconversion" identifier="2Dto3D"
id="1" export="false" split="false">
<option name="molgeo.algorithm" value="Distance geometry"/><option name="molgeo.tolerance" value="3"/>
</task>

<task name="SemiempiricalCalculation" identifier="MOPAC_OPT" id="2" export="false" split="true"
splitterTask="SplitStructureList" joinerTask="JoinStructureLists">
<option name="keywords" value="AM1 NOINTER MMOK
GNORM=0.1 EF"/>
</task>

<task name="SemiempiricalCalculation" identifier="MOPAC_PCalc" id="3" export="false" split="true"
splitterTask="SplitStructureList" joinerTask="JoinStructureLists">
<option name="keywords" value="AM1 VECTORS BONDS PI
POLAR PRECISE ENPART MMOK 1SCF"/>
</task>

<task name="DescriptorCalculation" identifier="DescCalc" id="4" export="false" split="false">
</task>

<task name="ModelBuilding" identifier="ModelBuild"
id="5" export="false" split="false">
<localInput source="H:\Unicore\test\Solub-data-water.plf" destination="SolubData"
type="http://www.openmolgrid.org/namespaces/PropertyFile"/>
</task>

<dependency pred="1" succ="2"/><!-- 2D-3D to MOP1 -->
<dependency pred="2" succ="3"/><!-- MOP1 to MOP2 -->
<dependency pred="3" succ="4"/><!-- MOP2 to DC -->
<dependency pred="4" succ="5"/><!-- DC to MB -->
</workflow>
```