1

OPEN COMPUTING GRID FOR MOLECULAR SCIENCES

Mathilde Romberg, Emilio Benfenati, and Werner Dubitzky

All substances are poisons; there is none that is not a poison. The right dose differentiates a poison from a remedy.

Paracelsus (1493–1541)

COPYRIGHTED MATERIAL

1.1 INTRODUCTION

The number of chemicals in society is largely increasing, and therewith the risk of being exposed to chemicals increases. Knowledge of possible toxic effects of these chemicals is vital, as are the measurement and assessment of the effects and related risks. Within the European Union, the Registration, Evaluation, and Authorisation of Chemicals (REACH) legislation [1] places responsibility on the chemical industries to properly assess the risks associated with their products. It has been estimated that about 30,000 new chemicals will be put on the European market in the coming years. The assessment of these chemicals would cost billions of euros and involve the use of millions of animals. REACH also aims to ensure that risks from substances of very high concern (SVHC) are properly controlled or that the substances are substituted. To match REACH requirements, fast and reliable methods with reproducible results are crucial, and regulatory bodies would be able to approve results. Property prediction and modeling will play an important role in this case [2].

Toxicology, the study of harmful interactions between chemicals and biological systems [3], uses more and more computer models. These models are based on already available data and help to reduce *in vivo* testing. Toxicity modeling and its data have many applications such as characterizing hazards, assessing environmental risks, and identifying potential lead components in drug discovery. A well-established method for toxicity modeling is quantitative structure–activity relationship (QSAR) or quantitative structure–property relationship (QSPR) [4,5]. On the basis of the available measured and calculated properties or activities and descriptors of compounds, predictive models for a certain property are built, which are then used to predict that

Grid Computing for Bioinformatics and Computational Biology. Edited by E.-G. Talbi and A.Y. Zomaya.
Copyright © 2008 John Wiley & Sons, Inc.

property for new compounds. An example for a property is the lethal dose (LD50), which is the amount of a substance that kills 50% of the population exposed to it. This property is mainly used to compare the toxicity of different compounds and to classify them, for example, for hazard warnings.

Classical QSAR models have been based on a very limited number of parameters, which have been measured (such as simple physicochemical properties) or calculated. The model target has been to find a relationship between these parameters and the property within a very limited congeneric series of chemicals. These chemicals share a common skeleton, and a few fragments are linked to it. In more recent years, there has been a significant change in the QSAR scenario: The interest has shifted from the identification of the relationship between the parameters and the property to a more practical use, the prediction of the properties of new chemicals. This calls attention to the predictive power of the model, since previously a model was not verified but was simply assessed with statistical measurements evaluating the fitting of the calculated values. Meanwhile, the challenge has become to model larger sets of compounds, and in addition the number of calculated chemical descriptors or fragments has drastically increased to several thousands. Finally, new more powerful algorithms are used, and these tools also introduce the possibility to extract new knowledge from the data instead of simply leading the algorithms toward well-known parameters based on *a priori* knowledge or hypotheses.

Classical bioinformatics applications such as data warehousing and data mining are a major part of the model development as a result of the following:

(a) The available data are stored in very different sources such as published journal papers, spreadsheets, and relational databases in different formats and notations with different nomenclature and

(b) Relations between the data are mined and used for building predictive models of various kinds such as multilinear regression (MLR), partial least squares (PLS), or artificial neural networks (ANNs).

Other applications applied within the process of prediction model development belong to the field of molecular modeling. Calculating certain properties of a molecule on the basis of its two- and three-dimensional structures provides the basis for the prediction of an endpoint such as LD50. Currently, the model-building and prediction process includes a variety of steps that a toxicologist or a pharmacologist would perform manually step by step by taking care of data selection, parameter setting, data format conversions and data transfer between each pair of subsequent steps, and so on.

Pharmaceutical industry and regulatory bodies together with environmental agencies are very interested in finding fast, cost-effective, easy, and reliable ways to identify compounds with respect to their toxicity. The process of determining lead compounds for a new drug takes years [6,7] in the laboratory, and in addition about 90% of the potential drugs entering the preclinical phase fail in further process due to their toxicity [8,9]. In recent years, pharmaceutical companies along with research initiatives have investigated modeling and prediction methods together with grid computing to

streamline and speed up processes. The prominent interest of industries lies in cost reduction, for example, reducing failure rate and using in-house PCs' idle time to run modeling tasks [10,11]. Software providers offer matching grid solutions [12,13] for the latter. These approaches exploit the embarrassingly parallel[1] nature of the applications and offer sophisticated scheduling mechanisms. They are deployed as in-house systems, that is, they do not span multiple organizations, mainly for security reasons. Companies do not risk their data and methods being exposed to outsiders.

Publicly funded research projects in bioinformatics investigate data and computational grid methods to integrate huge amounts of data, develop ontologies, model workflows, efficiently integrate application software including legacy codes, define standards, and offer easy-to-use and efficient tools [14–21].

Section 1.2 of this chapter will highlight grid systems in toxicology and drug discovery and their main characteristics. Section 1.3 will give an in-depth overview of the European OpenMolGRID approach, while Section 1.4 will conclude with an outlook for future developments.

1.2 GRIDS FOR TOXICOLOGY AND DRUG DISCOVERY

Toxicology covers important issues in life and environmental sciences. It is essential that the characteristics of a chemical be identified before producing and releasing it into the environment. In drug discovery, one aim is to exclude toxic, chemically unstable and biologically inactive compounds from the drug discovery [22,23] early on in the process. Therefore, models are being developed for predicting which compounds are liable to fail at a later stage of the process. In this context, QSAR models are one of the most popular methods. Another goal is to identify compounds that would bind to a given biological receptor. The area of docking is important to understand biological processes and find cures that succeed by activating or by inhibiting protein actions [24]. The docking studies require the modeling of the enzyme (which has to be known) in addition to the modeling of the small chemical compounds to be studied (ligand). However, these docking studies are more complex and do require a careful tridimensional description of the ligand. This is not always necessary in the case of QSAR models. For this reason, faster and simpler screening based on easier methods is often performed by drug companies, and the detailed docking studies are performed only for a limited number of chemicals. However, grid technologies introduce new possibilities.

The major objectives for using grid technology [25,26] to support biomathematics and bioinformatics approaches in drug discovery and related fields are to shorten the time to solution and reduce its costs. Users of such technology are (computational) biologists, pharmacologists, and chemists, who are usually not computer system experts. To bridge this gap, providing a user-friendly system is crucial. It allows

[1] An application is called embarrassingly parallel if no particular effort is needed to split it into a large number of independent problems that can be executed in parallel, and these processes do not need to communicate with each other.

the user to solve the biochemical problem without the knowledge about the details of the underlying system. Users require access to their private and publicly available data, execution of legacy software, and visualization of results. In cases where users develop their own application software, it should also be easily integratable.

There exist quite a few initiatives, projects, and systems that exploit grid methods for chemoinformatics, bioinformatics, and computational biology, and some of which focus on applications relevant to this chapter. One of the early grid projects in drug discovery is the Virtual Laboratory [27]. In the beginning of this century, the Virtual Laboratory project set up an infrastructure based on the Globus Toolkit 2.4 [28] and the Nimrod-G resource broker [29], specifically designed for parametric studies. The Virtual Laboratory environment provides software tools and resource brokers that facilitate large-scale molecular studies on geographically distributed computational and data grid resources. This is used for examining or screening millions of chemical compounds (molecules) in the Chemical Data Bank (CDB) to identify those having potential use in drug design. The DOCK software package [30] is integrated for molecular docking calculations and for access to the CDB, and the data replica catalogs are provided. The user interface allows us to specify input and output data sets and locations, set parameters for the DOCK software, submit jobs, and retrieve output. This command-line interface has recently been replaced by a Web portal within the Australian BioGrid initiative [31].

DDGrid [32] is a subproject of the China grid initiative that also focuses on docking. Its goal is to analyze chemical databases and identify compounds appropriate for a given biological receptor. It offers access to a variety of databases such as Specs (chemically available compounds' structure database), MDL Comprehensive Medicinal Chemistry 3D (pharmaceutical compounds), National Cancer Institute Database (NCI)-3D (structures with corresponding 3D models), China Natural Products Database, Traditional Chinese Medicinal Database (TCMD), and ZINC-ChemBridge (chemical structures). The user is provided with tools for preprocessing of data (Combimark), for visualization, for structure search, and for encrypting and decrypting of data. The core middleware layer is based on the Berkeley Open Infrastructure for Network Computing (BOINC, [33]), which is a well-established base for the group of "at home" systems. For example, Rosetta@home [34] uses PCs all over the world to model protein structures and interactions to understand diseases and find cures. Rosetta@home distinguishes itself from other grid initiatives in drug discovery by using voluntarily donated free CPU cycles to execute the Rosetta program. The "users" of these systems have no influence on the application. They download the software that uses the free CPU cycles to run the application set up by a research group, which in this case was David Baker's group at the University of Washington in Seattle [35].

Within the UK e-Science program, the e-Science Solutions for the Analysis of Complex Systems (eXSys) project [36] studies drug discovery from the angle of interaction networks. It analyzes protein interaction networks to identify sets of proteins in a bacterium that, if they were inhibited, would destroy the bacterium but not affect its host organism. These proteins qualify as potential drug targets. eXSys tackles data access and integration issues by building local data sets for intracellular

metabolic or protein interaction networks from heterogeneous resources such as the Database of Interacting Proteins (DIP, [37]), the Kyoto Encyclopedia of Genes and Genomes (KEGG, [38]), the Swissprot protein databank [39], and publications. A project internal common data format is established in which all data are integrated. The necessary network analysis programs plus their integration as grid services are developed. A graphical user interface allows users to select interaction networks from a local database, analyze them, and visualize the results. myGrid [40] and OGSA-DAI/OGSA-DQP [41] are used for data access and analysis of various data sources. The myGrid infrastructure offers a workbench well suited for bioinformaticians. It includes workflow generation and enactment, as well as a variety of services for data integration such as knowledge annotation and verification of experiments (KAVE), semantic discovery (Feta), and life science identifier (LSID) services for data handling.

The aspects of workflows for the drug design pipeline are also dealt with in the Wide *In Silico* Docking On Malaria (WISDOM) data challenge [42], a project to challenge the infrastructure built by the Enabling grids for e-Science (EGEE) project [43]. WISDOM seeks to find new inhibitors for a family of proteins by using *in silico* docking on the grid. During the 6-week data challenge in mid-2005, a terabyte of data was produced using 80 CPU years. These data from over 40 million docked ligands are now being further analyzed.

While most of the drug-discovery-related grid projects and systems deal with simulations and modeling of docking ligands to proteins and the identification of protein functions, it is also important to identify and optimize lead molecules with the targeted therapeutic potential. Pharmaceutical companies set up in-house grid systems to also cover this aspect. Little is published about these grid systems, but information can be found in case studies of software vendors [10] and press releases [44–46]. Key to this approach is workflow modeling, semantic Web technologies, and data management.

The approaches described use a variety of grid middleware or infrastructure systems, including grid service and pregrid-service versions of the Globus Toolkit, gLite (basis of the EGEE test bed), Web services, network computing (desktop grid), myGrid, and the pregrid-service version of UNICORE. With respect to middleware, all further developments aim at a service-oriented architecture (SOA), whatever type of resource is being used. The Open grid Services Architecture (OGSA [47]) has been defined by the Open grid Forum [48] to achieve a standardized grid framework. For drug discovery, the topics workflow modeling, application integration, standards for data structures, metadata and ontologies, and data integration are equally important.

Recently, a series of activities has addressed both the issues of databases of chemical compounds used in the world and how to predict their environmental and toxic properties. The European Commission's Joint Research Centre is considering the strategic development of a general system to predict properties of industrial chemicals. The U.S. Environmental Protection Agency (EPA), which adopts predictive tools for the property predictions of chemicals for decades, is also enlarging its set of tools. The Danish EPA predicted properties of tens of thousands of chemicals using a set of software. All these initiatives show the deep interest in a more powerful approach

capable to cope with a problem that involves many programs, databases, and resources, which would surely benefit from an integrated strategy supported by grid.

The following section will detail these characteristics taking the OpenMolGRID system as an example.

1.3 EXAMPLE OpenMolGRID

The Open Computing grid for Molecular Sciences and Engineering (OpenMolGRID, [49]) system has been developed to support the lead component identification in drug discovery and designed in an open manner to be of immediate use for other applications in biochemistry and pharmacology. The objectives of this project are to

- Develop tools that permit end users to securely and seamlessly access, integrate, and use relevant globally distributed data resources;
- Develop tools that permit end users to securely and seamlessly access, use, and schedule globally distributed computational methods and tools; and
- Provide foundations and design principles for developing and constructing next-generation molecular engineering systems.

The selected underlying grid middleware UNICORE [50] offers well-designed interfaces for application software integration both on the user client side and on the execution server side. It provides data access and data transfer together with workflow modeling, execution, and monitoring. To facilitate the development of prediction model and prediction workflows, the OpenMolGRID project developed abstraction layers to easily integrate application software and to access different publicly available relevant databases (e.g., ECOTOX [51], NTP [52]), and built a data warehouse from that data [53]. It includes automated workflow support that simplifies the task of the user by including support steps such as data conversion and data transfer into the workflow, by automatically assigning appropriate execution servers, and by exploiting parallelism [54,55].

The general architecture underlying the OpenMolGRID system is depicted in Fig. 1.1. The abstract resource interfaces are the key to flexibility providing a common interface accessible by the UNICORE server and a resource-specific interface on the resource side. Each resource has an XML resource description attached to inform the server and the client about input and output characteristics and behavior. The challenges addressed in the OpenMolGRID project are as follows:

- Molecular design and engineering are computationally very demanding, and they generates huge amounts of data.
- Data from a variety of different sources need to be integrated using data warehousing techniques, and the data need to be accessible seamlessly.
- The scientific workflows involve heterogeneous data, compute, and application resources.

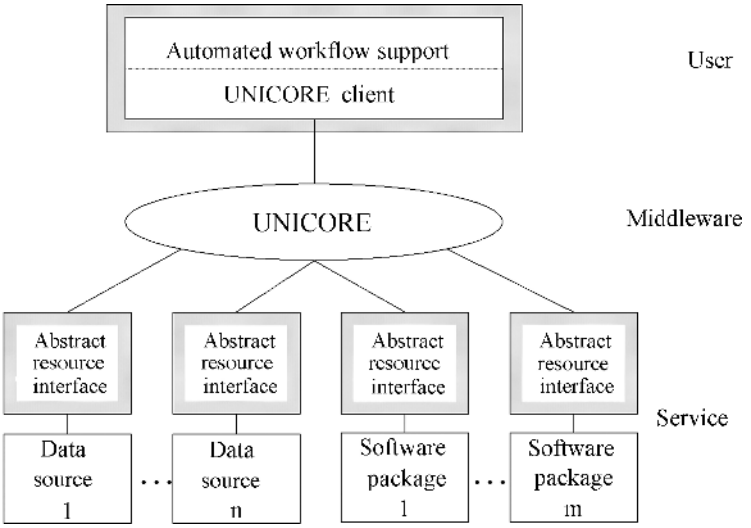

Figure 1.1. OpenMolGRID high-level architecture.

- The scientific workflows are fairly complex and involve multiple dependent steps.

These challenges become obvious when analyzing, for example, the QSAR model-building process: First, relevant experimental (toxicity) data are searched from one or multiple sources, which requires that all sources be accessible through the system and that the relevant data can be extracted. The structural information is then extracted and a 3D structure is generated for each molecule. Application software such as MOLGEO [56] is used to accomplish this task. In the next step, the generated molecular geometry is optimized using the molecular modeling software for example, MOPAC [57]. These applications process a single structure at a time so that the tasks can easily be distributed. This is also true for the next step in the workflow, the calculation of descriptors, for example, using the molecular descriptor analyzer (MDA) module from the Codessa [58] software package. The results from all structures serve as input to model development software—for example, the best multilinear regression (BMLR) analysis module of Codessa. The following sections will describe the OpenMolGRID solutions in detail.

1.3.1 Data Management

Besides storage of data, the major challenge in data management is the access to every kind of data source containing data relevant to drug discovery, toxicology, pharmacology, and, of course, the interpretation and integration of these data.

1.3.1.1 Access to Data Sources

The abstraction layer for data sources is realized as a set of metadata and a server-side wrapper application called database access tool that encapsulates the communication with the underlying database system. The output data are sent to the client in an XML format that is designed for easy automatic processing. The metadata file contains information on the database layout and semantic information—for example, database name, access restrictions, information about the database intended for the user, table names, field names, and types.

The XML structure defined to transfer the data from the data source to the user client or as input to other applications is a list of elements. This facilitates an easy transformation to other formats, for instance, application software input formats, and for easy extraction of certain fields, for example, the structure of a chemical compound.

1.3.1.2 Data Warehouse

Predictive QSAR/QSPR modeling requires the handling and management of chemical structure and property data, along with data relating to molecular descriptors. Often these data must be retrieved from public or private data repositories as well as integrated and formatted so that it is amenable to data mining methods such as linear regression methods, artificial neural networks, and decision tree algorithms. Data warehousing [59] provides the data integration and formatting functionality required by data mining applications. It is employed to integrate, cleanse, normalize, and consolidate data from different sources and to map them onto "ready-to-use" data structures (e.g., by denormalizing relational database tables). Within the OpenMolGRID system, a grid-enabled data warehouse for molecular engineering environments has been developed [53]. Its main purpose is to provide integrated and consolidated data originating from selected data resources relevant to molecular engineering. The following data resources have been integrated:

- National Toxicology Program database, which provides information regarding potentially toxic chemicals to health regulatory and research agencies, the scientific and medical communities, and the public NTP [52]
- ECOTOX (ecotoxicology) databases Aquire and Terretox, which provide chemical toxicity information for aquatic and terrestrial life, respectively [51]
- Multidrug resistance (MDR) data set, proprietary
- G-protein-coupled receptor (GPCR) data set, proprietary

The databases integrated in the data warehouse are harvested from the sources mentioned above and are mapped into the warehouse and its physical repository. A detailed view of the OpenMolGRID data warehouse and its relation to the Web and other OpenMolGRID components is depicted in Fig. 1.2. The warehouse processes follow the typical extract, transform, and load scheme (also known as ETL). According to reflect updates in the underlying databases, the warehousing process is performed periodically. The extract component transfers the database from its public Web site as single or multiple files (depending on the database) to the data warehouse. Each

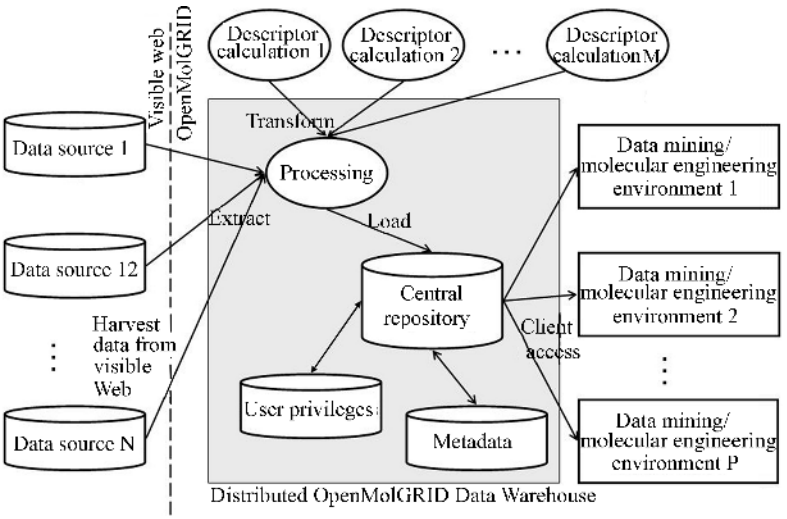

Figure 1.2. OpenMolGRID data warehouse and related components.

database has its own implementation-specific format from which the data are extracted and mapped into the data transformation environment. Within this environment, the data are denormalized from relational databases, cleansed (inconsistent entries are removed), enriched, and standardized based on the requirements of the molecular engineering environments. New fields are computed to facilitate different types of analyses. For example, the log inverse of the measured dosage of a chemical's toxicity is often more useful for some calculations and models than that of the toxicity value itself. By providing this value within the warehouse's data structures, the user does not have to perform this calculation and can focus on the more intricate aspects of the modeling task at hand. Data normalization may involve, for example, missing value imputation, mean centering, or alignment to canonical units. In addition, complex data transformations (descriptor calculations) are integrated into the warehouse again to offer values the users would have to calculate otherwise.

The transformed data are then loaded into the data warehouse's physical data storage, a PostgreSQL relational database. Client access to data in the OpenMolGRID data warehouse is enabled via the generic database access tool mentioned above. Inputs and outputs are encapsulated in an OpenMolGRID-specific XML syntax and data are easily identifiable due to being associated with generic data types defined especially for OpenMolGRID's data needs. These data types are used throughout all applications in the OpenMolGRID system.

The data warehouse's transformation environment includes the calculation of certain descriptor values as mentioned. Specialized software is required to perform these calculations and they are expensive to compute, especially if there are a large number of chemicals and several representations of the same chemical. From a data warehouse perspective, these descriptor calculations are complex data transformations.

Therefore, the most frequently used molecular descriptors are calculated for each molecular structure in the data warehouse. Besides the traditional molecular descriptor types, a physicochemical parameter, the log *P* value (octanol–water partition coefficient), is also calculated for the compounds. Essentially, the descriptor calculation procedure amounts to virtualization of parts of the data warehouse's data transformation processes. This virtualization functionality is realized by the development of the command-line interface for UNICORE.

1.3.1.3 A Data Type for Toxicity

A major challenge is that many data resources contain inconsistencies by way of the same data (types) represented in different records (the idea of a record varies from source to source). For example, supposing we have decided that the standardized data unit for a particular dosage field is milligrams per kilogram (mg/kg), there may be variations in the style, a source represents this. Some records may contain Mg/kg (or some other variation), thereby causing inconsistencies with the standard realized in the OpenMolGRID warehouse. The taxonomy step in the process flow described enables any number of substitutions to be defined to ensure that consistency is maintained within, and between, data resources entering the warehouse. Characteristic of many data sources is the idea that each data field contains a value from a set of allowable values. This can be problematic when a number of resources are being integrated into a data warehouse. For example, a dosage field can have several measurement units associated with it—for example, g/kg, mg/kg, or μg/kg—which have to be aligned to be usable for data mining. In the absence of a data warehouse, this must be performed manually, but in the OpenMolGRID data warehouse, anautomated mechanism is required. The mechanism was developed based on canonical units or primitives. Measurement units can be broken down into several categories—for example, length, weight, and time. Each of these categories has an associated base unit, the unit primitive—for example, kilograms for weight. For the conversion between various forms of the same measurement category, scaling factors (which can be more complex mathematical formulations) are defined, in both directions, to enable dynamic conversion from one unit to another.

1.3.1.4 Data Storage

Besides the data warehouse that contains the cleaned and transformed data from available data sources, space to store (intermediate) results is required. A relational database has been set up to support the complete molecular engineering process. It is capable of handling all data generated in the OpenMolGRID system (molecules, descriptors, models, experimental property values, predicted property values, etc.). It is set up as a read/write store, while the data warehouse is read-only from the end users' perspective.

Very important for a data store for molecular sciences is a structure and substructure search capability that has been developed. This function is necessary for identifying the best subset of data (chemical compounds) to be used for further analysis and is fundamental in chemical and related communities. The substructure search is realized

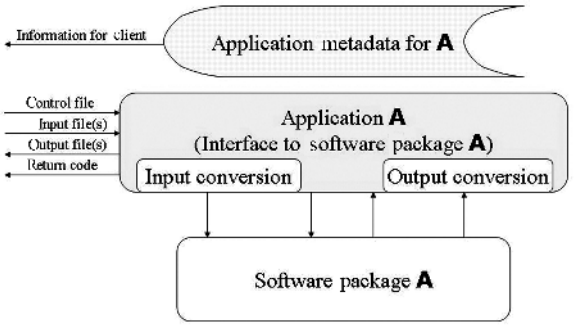

Figure 1.3. Application interface.

as a two-part process with two different queries. The first query aims to select a subset of structures that may contain the substructure. This is performed using a fingerprinting approach, which significantly accelerates the search. Fingerprints of the structures are matched against the fingerprint of the substructure, and those structures that cannot possibly match are removed from the set. The second query is performed within the matching subset to select structures that actually contain the full chemical substructure. The comparison is computationally expensive, which makes it important to use the first step to reduce the input set for the second query and thereby make the overall process more efficient.

1.3.2 Application Integration

Similar to the integration of data sources through metadata and access software (database access tool), all kinds of applications can be integrated into the system as shown in Fig. 1.3. The abstract interface is realized as a wrapper to existing software modules. It provides the description of the application (its metadata) and the input/output (I/O) data format conversion routines from the standard data format to the proprietary and vice versa. The metadata also define the interface and I/O format information used by clients. As a result, a well-defined application on the server side can be addressed on the user client side by an application-specific interface as shown in the following section.

1.3.3 User Interface

OpenMolGRID, being based on the UNICORE grid middleware [60], includes the UNICORE graphical client (see Fig. 1.4), which is shown here with the detailed workflow for model building (for the description of the coarse-grained workflow, see introduction to Section 1.3). It is a Java application offering job creation and monitoring for complex multistep and multisite jobs. Jobs are composed of subjobs, tasks, and dependencies reflecting temporal and data dependencies. Jobs are represented by acyclic directed graphs with tasks and subjobs as vertices and dependencies

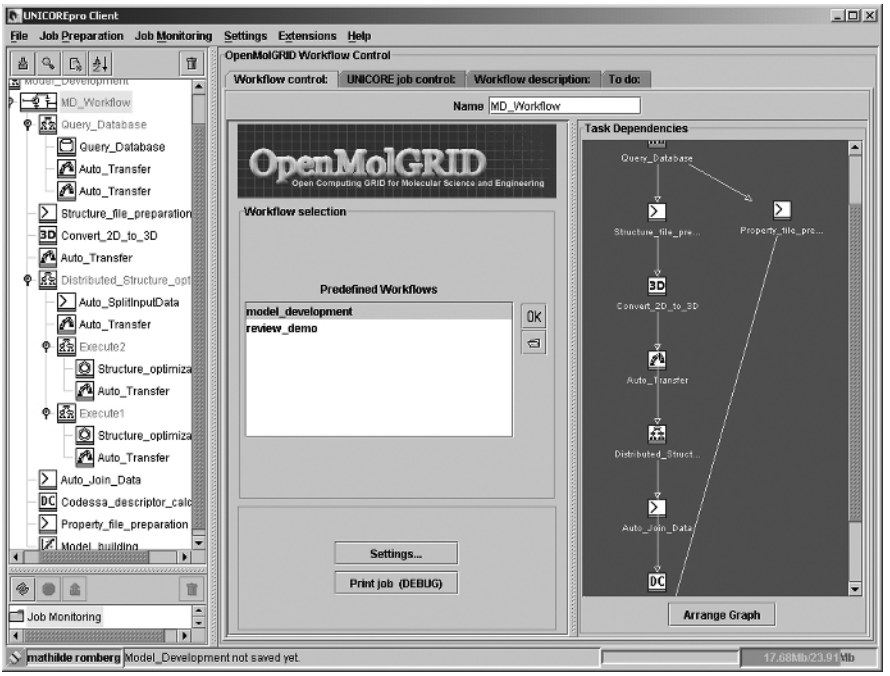

Figure 1.4. User client.

as edges. UNICORE jobs representing subjobs are associated with a target system (vsite), where the (part of the) job should be executed. The basic tasks the client provides are *import* of input data files, *export* of output data files, *transfer* of data between subjobs, *execute* of a program, and dependencies for simple sequential dependency, *if-then-else*, *while*, *repeat*, and *hold*. The most important feature in the OpenMolGRID context is the client's plugin interface to integrate application-specific plugins that represent new flavors of the execute task. The application-specific plugins correspond to defined application resources on the server side as described in Section 1.3.2.

1.3.4 Workflow Modeling

The specification and execution of complex workflows such as those in molecular design and engineering using grid resources is still under research [61]. Workflow solutions exist mostly for business processes. Languages to describe business processes are, for example, BPEL4WS (Business Process Execution Language for Web Services, see Reference 62) and WPDL (Workflow Process Definition Language; see Reference 63). The modeling of complex workflows in the scientific arena is mostly performed manually using the tools the existing grid middleware offers [64,65]. The key to automated generation of workflows is the description of software resources available on grid computing resources. In the OpenMolGRID system, this is provided

by the application's abstraction layer. These descriptions can be used for automated application identification and inclusion in multistep workflows. The following paragraphs will describe the solution for automated workflow specification and processing developed within the OpenMolGRID project.

Workflow Specification. Workflows consist of task or process elements with logical, temporal, and data dependencies. Tasks may also be independent of other tasks in a workflow. As described in Section 1.3.3, OpenMolGRID uses the UNICORE client that offers graphical workflow specification to build up complex jobs. Existing workflow description languages do not match the UNICORE model with respect to application software resources. As these play the most important role within the automatic job generation, a workflow specification language has been developed that enables 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, as well as the necessary resources. XML has been selected as a specification language for the workflow. A core element in a workflow is the task, which is described by

- A string, the task's function to be fulfilled by an application resource and supported by a client plugin
- A string, the UNICORE task identifier in the job tree
- An integer, the unique numerical task identification within the workflow
- A boolean, the export flag specifying whether result files are to be exported to the user's workstation
- A boolean, the split flag specifying whether the task is data parallel and can be distributed onto several execution systems
- A string, the identifier of an application that is capable of splitting the input data for this task into *n* chunks (splittertask)
- A string, the identifier of an application that is capable of joining the *n* result files into one file (joinerTask)
- A set of options to feed the application with parameter settings

A set of resources is specified for a task requesting runtime, number of nodes, number of processors, and memory. The target system for the execution of all tasks within a UNICORE (sub)job can be specified by usite and vsite.

The following shows the XML workflow specification for the model development process:

```
<?xml version="1.0" ?>
<!– *** Model Multi-Drug resistance on OpenMolGRID data warehouse data ***
–– >
<workflow xmlns="http://www.openmolgrid.org/namespaces/2004/Workflow
  Description" xmlns:rd="http://www.openmolgrid.org/namespaces/2004/
  SimpleResources" >
   <group type="subjob" identifier="Query Database" id="1" >
```

```
<!– wrapper group to allow easy datasource selection –>
  <task name="DataBaseRequest" identifier="Query Database" id="11"
    export="false" split="false" >
    <option name="query" value="SELECT chemical.moldw_id,
    chemical.structuretype, chemical.fileformat,chemical.
    molecularstructure,property.propertyid,property.propertyname,
    property.loginverse FROM (chemical LEFT JOIN property ON
    chemical.moldw_id=property.moldw_id) WHERE chemical.
    molecularstructure!=" and property.propertyname like 'Multi-Drug%'" />
  </task>
  <task name="DataBaseRequestToPLF" identifier="Property file
    preparation" id="3" export="false" split="false" />
  <task name="DataBaseRequestToSLF" identifier="Structure file
    preparation" id="2" export="false" split="false" />
  <resourceRequest>
    <rd:node usite="Ulster OMG" vsite="MOLDW" />
  </resourceRequest>
</group>
<task name="2Dto3Dconversion" identifier="Convert 2D to 3D" id="21"
  export="false" split="false" />
<task name="SemiempiricalCalculation" identifier="Structure optimization"
  id="25" export="false" split="true" splitterTask="SplitStructureList"
  joinerTask="JoinStructureLists" >
  <option name="keywords" value="AM1 PRECISE 1SCF NOINTER" />
</task>
<task name="DescriptorCalculation" identifier="Codessa descriptor
  calculation" id="29" export="false" split="false" />
<task name="ModelBuilding" identifier="Model building" id="40"
  export="false" split="false">
  <resourceRequest>
    <rd:resources runTime="3600" />
  </resourceRequest>
</task>
<dependency pred="11" succ="2" />
  <!– db request to structure extract –>
<dependency pred="11" succ="3" />
  <!– db request to property extract –>
<dependency pred="3" succ="40" />
  <!– property extract to model building –>
<dependency pred="2" succ="21" />
  <!– struct extract to 2d to 3d –>
<dependency pred="21" succ="25" />
  <!– 2d to 3d to semiempirical –>
<dependency pred="25" succ="29" />
  <!– semiempirical to descriptor calc –>
<dependency pred="3" succ="40" />
  <!– property extract to model building –>
<dependency pred="29" succ="40" />
  <!– descriptor calc to Modelbuilding –>
<resourceRequest>
  <rd:node usite = "TartuOMG" vsite = "VSite1"/ >
</resourceRequest>
</workflow>
```

Workflow Processing. A workflow specified as described serves as input to the MetaPlugin, a special plugin to the UNICORE client. The MetaPlugin parses the XML workflow, creates the corresponding UNICORE job, and assigns target systems and resources to it. These tasks include a number of sophisticated actions:

- Subjobs are introduced into the job wherever necessary, for example, when requested applications are not available on the same target system.
- Transfer tasks are introduced into the job to transmit data from one target system to another, which is the execution system of a subjob.
- Data conversion tasks are 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 are added to the workflow as predecessor tasks of a splitable task for input data preparation.
- Subjobs are created around split tasks for each selected target system and a transfer task to transfer the output data back to the superordinate subjob.
- Joiner tasks are 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) is 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 to the client. A resource information provider component has been developed to support the MetaPlugin in resource selection. It returns the client plugin handling the function, the target systems offering the corresponding application, and the I/O formats. Currently, the MetaPlugin does resource selection at a basic level, but a more sophisticated resource broker component can 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 complex workflow of the corresponding UNICORE job. The latter would demand detailed knowledge about, for example, I/O formats for the inclusion of data conversion tasks and the manual splitting and distribution of tasks onto appropriate target systems. The automatic UNICORE job creating gives the flexibility to the system to adapt to the actual grid layout and resource availability and helps to avoid human errors.

Figure 1.4 shows on the left-hand side the UNICORE job the MetaPlugin generated from the workflow detailed. All tasks starting with "Auto_" have been added as are the groups "Execute1" and "Execute2," where the semiempirical calculation is distributed among systems offering the necessary application software. "Auto_Transfers" are used, for example, to transfer data from the database to the systems where the structure conversion and the model development are to be executed. The "Auto_SplitInputData" and "Auto_Join_Data" tasks have been included to partition the input data for the semiempirical calculation to allow for its distributed execution and to join the result files after the execution is completed.

1.3.5 Experience

One of the most noteworthy outcomes of the OpenMolGRID project is that it paves the way for standardization of model-building and prediction processes. Within OpenMolGRID, we checked that results obtained with the automatic process are equivalent to those obtained by manual modeling, both for the chemical descriptor calculation and for the QSAR results. This shows that OpenMolGRID has practical potential as a regular tool for QSAR modeling. It is important to note the advantages in this direction offered by the OpenMolGRID approach for models for regulatory purposes. Indeed, in the case of scientific applications the automatic simplified process is surely convenient and appealing, and it speeds up the application. But for regulatory purposes, it becomes necessary to get the same result independently of the user. So far in most of the cases QSAR models, except those within the classical approach with a few simple parameters, require manual steps that produce variable results due to optimization differences and the lack of sufficient details. The availability of automatic QSAR modeling tools would surely cover the need for more reproducible results, which is a requirement for results to be used within a regulatory context: The implementation of a candidate protocol for QSAR modeling as a workflow would achieve reproducibility, easy models, and suitability for regulatory purposes [66].

Mazzatorta et al. [67] obtained stable and thoroughly validated QSARs using the OpenMolGRID system, and Maran et al. [68] detailed the use of OpenMolGRID for the development of QSAR models for the prediction of HIV-1 protease inhibition ability of potential inhibitors. They pointed out that building the model is accomplishable within 1 h using the system instead of 1 day because of automation of the workflow and parallel execution of tasks. This shows that the objective to shorten the time to solution has been achieved. Especially the automatic distribution of a task onto the available systems and the automated output/input format conversion account for this.

During the development of the system, application integration has proved to be easy because the application software itself does not need to be adapted, only a wrapper implementing the abstract interface has to be developed. The data warehouse's transformation process has been significantly improved by the provision of a command-line interface and a queuing component to the UNICORE system. The data warehouse uses these components to submit descriptor calculation to grid resources and retrieve the output. Indirectly, this also speeds up the user's workflows because values that every user requires are already calculated up-front and provided in the data warehouse.

The current lack of an XML editor to generate the workflows makes it difficult for the toxicologists to prepare their own workflows. These have to be prepared by someone familiar with XML and the workflow schema which can easily be covered for standard workflows that are prepared initially and made available to everyone—for example, the model development process, but not for, for example, experimental workflows.

A set of open issues has arisen from data handling. Standard formats should be used wherever possible, but, for example, there is not yet an established standard

for globally unique identifiers for molecular descriptors. How to store the predictive models has not yet been resolved because PMML (predictive model markup language, [69]) is not sufficient, and it cannot be used to describe PLS (partial least square) or PCR (principal components regression) models. An extension to PMML could be a solution. For a molecular structure, multiple conformations can exist, and the handling of these multiple data including their storage, selection, and processing has not yet been resolved in chemoinformatics. These topics will, among others, be dealt with in the EC-funded project Chemomentum [70].

1.4 SUMMARY AND OUTLOOK

QSAR modeling, *in silico* modeling, is a prominent method in toxicology and pharmacology. Important is the quality of the models; for example, the REACH legislation requires a clear estimation of the quality of a model, its accuracy, before it can be used for REACH purposes. The European Chemical Bureau is coordinating an action on QSAR [71] in support of prediction modeling for regulatory purposes. Within drug discovery, it is vital to have significant models for the prediction of ADMET (absorption, distribution, metabolism, excretion, and toxicity) properties [72]. The mathematical modeling is supported by grid computing because it helps speed up the process of model building through parallelization and user-friendliness. Pharmaceutical companies save further costs with in-house grids using their idle desktop PC cycles instead of investing in additional compute power. The emerging service-oriented architectured grid systems allow interoperability and thereby support for true global sharing of all kinds of resources (CPU cycles, data, knowledge, applications, etc.). This enables collaboration and may lead to synergy in achieving better and quicker results. Data and knowledge management are key to interoperability. The data from different sources need to be interpretable, requiring ontologies to be further developed to enable "understanding." The Semantic grid [73] offers the necessary framework. It can be used to automate the integration of data from different sources and their transformation into knowledge. The Comb*e*Chem UK e-Science project [74] demonstrates the advantages of Semantic grid for data from chemistry. On the basis of a chemistry ontology and RDF (Resource Description Framework) graphs using XML data descriptions, it provides a flexible data structure for data integration and knowledge management. Drug development and toxicology are going to gain from the smart laboratory that has developed. An aspect to be covered is data privacy and security, especially for company-owned data or medical records. These data would not be allowed to leave the source, but it may be allowed to mine the data locally and transfer only the results. Therefore, distributed data mining is another topic needing further research [75]. In addition to security issues, it may not be feasible to transfer data to an execution server because of their sheer volume.

In the future, grid computing will further impact procedures in toxicology and pharmacology. Having standard procedures will help regulatory bodies in their decisions. High-quality prediction models will reduce the amount of *in vitro* and *in vivo* testings. Nanotoxicology [76], the research on toxic effects of nanoparticles (particles

of nanometer size), will need computational and prediction models and procedures for determining physicochemical parameters and effects and for risk assessment. Drug development will extend its use of computational methods and knowledge exploitation to find cures. The systems, biology approach to simulate all processes in a biological system—for example, a cell—is used to further understand the way the system works and can be influenced, which may improve drug discovery [77].

ACKNOWLEDGMENTS

This work has partially been funded by the European Commission under Grants IST-2001-37238 and FP6-2005-IST-5-033437.

REFERENCES

1. EU Chemicals Legislation—REACH (Registration, Evaluation and Authorisation of Chemicals). http://ec.europa.eu/enterprise/reach/index_en.htm.
2. M.T.D. Cronin, J.S. Jaworska, J.D. Walker, M.H.I. Comber, C.D. Watts, A.P. Worth, Use of QSARs in international decision-making frameworks to predict health effects of chemical substances, *Environmental Health Perspectives* **111**(10), 1391–1401, 2003.
3. J. Timbrell, *Introduction to Toxicology*, third edition, Taylor & Francis, Boca Raton, FL, 2002.
4. C. Hansch, A. Leo, Exploring QSAR: *Fundamentals and Applications in Chemistry and Biology*, Vol. 1, American Chemical Society, Washington, DC, Chapters 6 and 11, 1995.
5. J.D. McKinney, A. Richard, C. Waller, M.C. Newman, F. Gerberick, The practice of structure activity relationships (SAR) in toxicology, *Toxicological Sciences* **56**(1), 8–17, 2000.
6. American Association for the Advancement of Science. http://www.aaas.org/international/africa/gbdi/mod1c.html.
7. PPD Corporate Information. http://www.ppdi.com/corporate/faq/about_drug_development/home.htm.
8. UK Department of Trade and Industry (DTI), call for research proposals. http://www.technologyprogramme.org.uk/site/DTISpring06/Spring06Docs/BioscienceHealthcare06.pdf.
9. J.A. DiMasi, Risks in new drug development: Approval success rates for investigational drugs, *Clinical Pharmacology & Therapeutics* **69**(5), 297–307, 2001.
10. Novartis Grid MP case study. http://www.ud.com/resources/files/cs_novartis.pdf.
11. ADMEToxGrid project. http://www.lpds.sztaki.hu/index.php?loadp̄rojects/current/comgenex.php.
12. United Devices Grid MP on Intel Architecture. http://www.ud.com/resources/files/wp_intel_ud.pdf.
13. Platform Computing products for life sciences. http://www.platform.com/Life.Sciences/CustomersCaseStudies.htm.
14. North Carolina bioportal. http://www.ncbiogrid.org.
15. Workflow management for bioinformatics (pegasys). http://bioinformatics.ucb.ca/pegasys.
16. Cancer biomedical informatics grid (cagrid). https://cabig.nci.nih.gov/workspaces/Architecture/caGrid.
17. BioPAUÁ. http://www.biopaua.lncc.br/ENGL/index.php.
18. Open bioinformatics grid. http://www.obigrid.org.
19. Asia Pacific biogrid initiative. http://www.apbionet.org/apbiogrid.
20. Genegrid—virtual bioinformatics laboratory. http://www.qub.ac.uk/escience/dev/article.php, section projects.
21. OpenMolGrid project. http://www.openmolgrid.org/.

22. Drug Discovery Process, graphics by Novartis. http://www.nibr.novartis.com/images/OurScience/drug.discovery_graph.jpg.
23. Genelabs' Drug Discovery Process. http://www.genelabs.com/research/discoveryProcess.html.
24. E.A. Lunney, Computing in drug discovery: The design phase, *IEEE Computing in Science and Engineering* **3**(5), 105–108, 2001.
25. I. Foster, C. Kesselman (Eds.), *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, San Francisco, CA, 1998.
26. I. Foster, C. Kesselman (Eds.), *The Grid 2: Blueprint for a New Computing Infrastructure*, second edition, Morgan Kaufmann Publishers, San Francisco, CA, 2004.
27. R. Buyya, K. Branson, J. Giddy, D. Abramson, The virtual laboratory: enabling molecular modelling for drug design on the World Wide Grid, *Concurrency and Computation: Practice and Experience* **15**(1), 1–25, 2003.
28. Globus Toolkit 2.4 Overview. http://www.globus.org/toolkit/docs/2.4/overview.html.
29. R. Buyya, D. Abramson, J. Giddy, Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid, in: *Proceedings of the HPC ASIA 2000, Beijing, China*, IEEE Computer Society Press, Los Alamitos, CA, 2000.
30. B. Shoichet, D. Bodian, I. Kuntz, Molecular docking using shape descriptors, *Journal of Computational Chemistry* **13**(3), 380–397, 1992.
31. H. Gibbins, K. Nadiminti, B. Beeson, R. Chhabra, B. Smith, R. Buyya, The Australian BioGrid Portal: Empowering the Molecular Docking Research Community. Technical Report, GRIDS-TR-2005-9, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, June 13, 2005.
32. W. Zhang, J. Shen, Drug Discovery Grid (DDGrid), Second Grid@Asia Workshop, Shanghai, China, February 20–22, 2006.
33. Berkeley Open Infrastructure for Network Computing (BOINC). http://boinc.berkeley.edu/.
34. O. Schueler-Furman, C. Wang, P. Bradley, K. Misura, D. Baker, Progress in modeling of protein structures and interactions, *Science* **310**, 638–642, 2005.
35. The Baker Laboratory. http://depts.washington.edu/bakerpg/.
36. e-Science Solutions for the Analysis of Complex Systems (eXSys). http://www.neresc.ac.uk/projects/eXSys.
37. Database of Interacting Proteins (DIP). http://dip.doe-mbi.ucla.edu/.
38. KEGG: Kyoto Encyclopedia of Genes and Genomes. http://www.genome.ad.jp/kegg/.
39. UniProtKB/Swiss-Prot Protein Knowledgebase. http://www.ebi.ac.uk/swissprot/.
40. R. Stevens, A. Robinson, C.A. Goble, my Grid: personalised bioinformatics on the information grid, *Bioinformatics* **19**(Suppl. 1), i302–i304, 2003.
41. Open Grid Services Architecture—Data Access and Integration. http://www.ogsadai.org.uk/.
42. Initiative for grid-enabled drug discovery against neglected and emergent diseases. http://wisdom.eu-egee.fr/.
43. Enabling Grids for e-Science. http://www.eu-egee.org/.
44. Will LION and IBM succeed with the drug research Grid solution? http://www.gridtoday.com/02/1111/100708.html.
45. Platform partners with Matrix Science to accelerate drug discovery for pharmaceutical companies. http://www.hoise.com/primeur/02/articles/weekly/AE-PR-12-02-38.html.
46. Output of e-Science project helps GSK speed up drug discovery. http://www.rcuk.ac.uk/escience/news/ahm9.asp.
47. The Open Grid Services Architecture, Version 1.0. 2005. Global Grid Forum, http://www.gridforum.org/documents/GFD.30.pdf.
48. The Open Grid Forum (former Global Grid Forum and Enterprise Grid Alliance). http://www.ogf.org.

49. M. Romberg (Ed.), *OpenMolGRID—Open Computing Grid for Molecular Science and Engineering*, NIC series No. 29, Forschungszentrum Julich, ISBN 3-00-016007-8, July 2005.
50. UNICORE—Uniform Interface to Computing Resources. http://unicore.sourceforge.net.
51. ECOTOXicology Database. http://www.epa.gov/ecotox.
52. National Toxicology Program. http://ntp-server.niehs.nih.gov.
53. W. Dubitzky, D. McCourt, M. Galushka, M. Romberg, B. Schuller, Grid-enabled data warehousing for molecular engineering, *Parallel Computing* **30**(9–10), 1019–1035, 2004.
54. B. Schuller, M. Romberg, L. Kirtchakova, Application Driven Grid Developments in the OpenMolGRID Project, in: P.M.A. Sloot, A.G. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak (Eds.), *Advances in Grid Computing—EGC 2005*, LNCS 3470: February 23–29, 2005.
55. S. Sild, U. Maran, M. Romberg, B. Schuller, E. Benfenati, OpenMolGRID: Using automated workflows in GRID computing environment. in: P.M.A. Sloot, A.G. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak (Eds.), *Advances in Grid Computing—EGC 2005*, LNCS 3470, pp. 464–473, February 2005.
56. E.V. Gordeeva, A.R. Katritzky, Rapid conversion of molecular graphs to three-dimensional representation using the MOLGEO program, *Journal of Chemical Information and Computer Sciences* **33**, 102–111, 1993.
57. J.J. Stewart, MOPAC: A semiempirical molecular orbital program, *Journal of Computer-Aided Molecular Design* **4**, 1–45, 1990.
58. COmprehensive DEscriptors for Structural and Statistical Analysis (Codessa) QSPR/QSAR Software, http://www.codessa-pro.com/index.htm.
59. L. Moss, A. Adelman, Data warehousing methodology, *Journal of Data Warehousing* **5**, 23–31, 2000.
60. D. Erwin (Ed.), UNICORE Plus Final Report—Uniform Interface to Computing Recources, Joint Project Report for the BMBF Project UNICORE Plus Grant Number: 01 IR 001 A-D, ISBN 3-00-011592-7. Available at http://www.unicore.org/documents/UNICOREPlus-Final-Report.pdf.
61. Open Grid Forum (OGF) Research Group on Workflow Management (WFM-RG). https://forge.gridforum.org/projects/wfm-rg/.
62. Business Process Execution Language for Web Services. Version 1.0. July 31, 2002. http://www-106.ibm.com/developerworks/library/ws-bpel1/.
63. M. zur Muehlen, J. Becker, WPDL: State-of-the-art and directions of a meta-language for workflow processes, in: Lothar Bading, Boris Pettkoff, August-Wilhelm Scheer, Siegfried Wendt (Eds.), *Proceedings of the 1st Know-Tech Forum*, Potsdam, 1999.
64. D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M.R. Pocock, P. Li, T. Oinn, Taverna: A tool for building and running workflows of services, *Nucleic Acids Research* **34**(Web Server issue):W729–W732, doi:10.1093/nar/gkl320, 2006.
65. S. Majithia, M.S. Shields, I.J. Taylor, I. Wang, Triana: A graphical web service composition and execution toolkit, *In Proceedings of the IEEE International Conference on Web Services* (ICWS '04), IEEE Computer Society, Los Alamitos, CA, pp. 514–524, 2004.
66. E. Benfenati, Modelling Aquatic Toxicity with Advanced Computational Techniques: Procedures to Standardize Data and Compare Models, in: J.A. López, E. Benfenati, and W. Dubitzky (Eds.), *Proceedings of International Symposium Knowledge Exploration in Life Science Informatics 2004* (KELSI 2004), Lecture Notes in Computer Science, LNCS 3303, Springer, New York, pp. 235–248, 2004.
67. P. Mazzatorta, M. Smiesko, E. Lo Piparo, E. Benfenati, QSAR model for predicting pesticide aquatic toxicity, *Journal of Chemical Information Model*, **45**, 1767–774, 2005.
68. U. Maran, S. Sild, I. Kahn, K. Takkis, Mining of the chemical information in GRID environment, Future Generation Computer Systems, in press, Corrected Proof, Available online 5 July 2006.
69. Predictive Model Markup Language. http://www.dmg.org/pmml-v3-1.html.
70. Chemomentum (Grid Services based Environment to enable Innovative Research). http://www.chemomentum.org.

71. Computational Toxicology (including QSARs) Joint Research Centre (JRC) Action no1321 of the European Chemicals Bureau (ECB). http://ecb.jrc.it/QSAR/.

72. I.V. Tetko, P. Bruneau, H.-W. Mewes, D.C. Rohrer, G.I. Poda, Can we estimate the accuracy of ADME-Tox predictions? *Drug Discovery Today*, **11**(15–16), 700–707, 2006.

73. D. De Roure, N.R. Jennings, N.R. Shadbolt, The Semantic Grid: A Future e-Science Infrastructure, in: F. Berman, A.J.G. Hey, and G. Fox (Eds.), *Grid Computing: Making the Global Infrastructure a Reality,* John Wiley & Sons, Hoboken, NJ, 2003, pp. 437–470.

74. K.R. Taylor, J.W. Essex, J.G. Frey, H.R. Mills, G. Hughes, E.J. Zaluska, The semantic grid and chemistry: experiences with combechem, *Journal of Web Semantics* **4**(2), 84–101, 2006.

75. D.B. Skillicorn, Distributed Data-Intensive Computation and the Datacentric Grid, White Paper, School of Computing, Queen's University, Kingston, Canada, 2003.

76. A. Bassan, S. Eisenreich, B. Sokull-Kluettgen, A. Worth, The role of ECB in the risk assessment of nanomaterials. Is there a future for "computational nanotoxicology"? Poster, The 12th International Workshop on QSAR in Environmental Toxicology.
http://ecb.jrc.it/DOCUMENTS/QSAR/INFORMATION_SOURCES/
PRESENTATIONS/Bassan_Lyon_0605_poster.pdf.

77. E.C. Butcher, Can cell systems biology rescue drug discovery? *Nature Reviews Drug Discovery* **4**(6), 461–467, 2005.