Workflows for Metabolic Flux Analysis: Data Integration and Human Interaction

Tolga Dalman, Peter Droste, Michael Weitzel, Wolfgang Wiechert, and Katharina Nöh

Institute of Biotechnology 2, Forschungszentrum Jülich, 52425 Jülich, Germany
{t.dalman,p.droste,m.weitzel,w.wiechert,k.noeh}@fz-juelich.de

Abstract. Software frameworks implementing scientific workflow applications have become ubiquitous in many research fields. The most beneficial advantages of workflow-enabled applications involve automation of routine operations and distributed computing on heterogeneous systems. Particular challenges in scientific applications include grid-scale orchestration of complex tasks with interactive workflows and data management allowing for integration of heterogeneous data sets.

We present a workflow for the ^{13}C isotope-based Metabolic Flux Analysis (13C-MFA). The core of any 13C-MFA study is the *metabolic network modeling* workflow. It consists of sub-tasks involving model set-up and acquisition of measurement data sets within a graphical environment, the evaluation of the model equations and, finally, the visualization of data and simulation results. Human intervention and the integration of various knowledge and data sources is crucial in each step of the modeling workflow. A scientific workflow framework is presented that serves for organization and automation of complex analysis processes involved in 13C-MFA applications. By encapsulating technical details and avoiding recurrent issues, sources for errors are minimized, the evaluation procedure for ^{13}C labeling experiments is accelerated and, moreover, becomes documentable.

Keywords: Scientific Workflows, Human Tasks, Database Integration, 13C-MFA, SOA.

1 Introduction

In recent years, scientific workflows emerged as a key technology in a growing number of research fields. The most beneficial advantages of workflow-enabled scientific applications involve automation of routine operations and distributed computing on heterogeneous systems. Scientific workflow systems have been realized successfully in various research fields [13,7,11,14] and nowadays a plenitude of successful workflow applications are available [16,12,1]. The diversity of these examples show: scientific workflow frameworks are inherently domain-dependent.

Current challenges in the workflow field include Grid-scale orchestration of complex, long-running tasks with interactive workflows. We present an application in the field of Metabolic Engineering where all of these challenges need

T. Margaria and B. Steffen (Eds.): ISoLA 2010, Part I, LNCS 6415, pp. 261–275, 2010.

to be addressed. In particular, we show that domain-specific requirements for a scientific workflow framework managing Metabolic Engineering applications is of central importance.

1.1 Metabolic Flux Analysis with Labeling Experiments

The major objective in Systems Biology and Metabolic Engineering is to understand complex processes in biological organisms [10,18]. In these research fields a combination of huge amounts of diverse data, advanced analytical tools, and the use of mathematical models aims at a targeted *in silico* design of optimized cell factories.

The major functional determinants of cell physiology are the *in vivo* reaction rates, i.e. the velocity in which certain substances are transformed into another. Thus, the understanding of the microorganisms' metabolic pathways is the key for a directed cell engineering towards the next generation of, e.g., high-performance producing strains. The intracellular reaction rates, however, cannot be directly observed. Instead, the rates, or synonymously fluxes, have to be indirectly estimated by a model-based analysis of stable isotope patterns from carbon labeling experiments, called ^{13}C-Metabolic Flux Analysis (13C-MFA) [21,26,28].

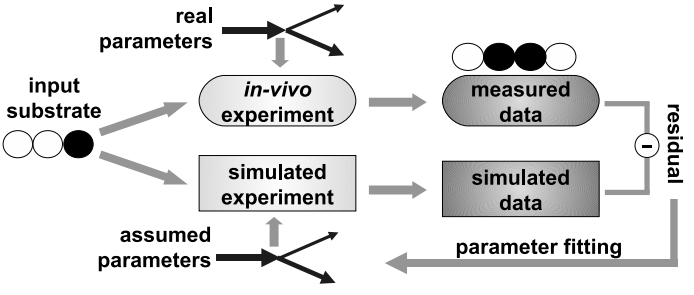

Fig. 1. Principle of Metabolic Flux Analysis with carbon labeling experiments

Current practice of 13C-MFA involves a variety of steps: post-processing of measured raw data from labeling experiments, computer-aided knowledge acquisition of metabolic pathways, the employment of sophisticated software tools for modeling and specially-tailored visualization for the interpretation of analysis results. A typical workflow can be sketched as follows:

1. In a carbon labeling experiment, living cells are fed with specifically ^{13}C-labeled substrates. The distribution of labeling patterns within intracellular metabolites is detected with highly sensitive mass spectrometry devices. Subsequently, these measurements are post-processed and collected in spread sheets.

2. Organism-specific information about metabolic reactions is acquired from publicly available resources and expert knowledge. Based on this information, a metabolic model is set up.
3. Mass balance equations are automatically generated that describe the metabolites' labeling patterns in detail. Various *in silico* methods are utilized to explore the capabilities of the model for different parameter settings.
4. By comparison of the experimentally acquired measurement data with the simulated measurement data, the real flux distribution is estimated from an iterative parameter estimation approach.

The 13C-MFA procedure shown in Fig. 1 is reviewed in more detail elsewhere [26,28]. In real world applications, this workflow is far from being a straightforward procedure, but involves many iterative cycles. Thus, a scientific workflow system organizing 13C-MFA processes in an automated fashion is needed.

1.2 Scientific Workflow Applications in the 13C-MFA Domain

The term *scientific workflow* emphasizes the extended needs of workflow applications in the research domain [22]. Contemporary topics closely related to Grid computing and scientific workflows include:

- *Workflow orchestration*: a scientific workflow engine is responsible for the invocation and distribution of tasks onto (compute) resources. In the scientific community, Taverna [7] and Kepler [13] are widely known workflow engines.
- *Grid provisioning*: on-demand acquisition of Grid resources is a desirable feature of a scientific workflow framework. Various applications address the integration of resource provisioning into workflow frameworks [2,14,24].
- *Data management*: scientific applications often require the handling of large amounts of data. The integration of scientific data into workflow and Grid frameworks has been presented in various projects [19,1].

Each scientific domain imposes specific requirements onto a scientific workflow framework. These non-standard requirements for workflow-enabled 13C-MFA applications have been identified:

(a) *Human interaction*: 13C-MFA workflows typically involve manual tasks consciously performed by human experts. The scientific workflow framework needs to be capable of modeling and integrating human actor steps in an appropriate way.
(b) *Dynamic workflows*: because typical 13C-MFA workflows consist of several steps interconnected by conditional branches and loops, a *control-based* workflow modeling scheme is preferred to solutions with an emphasis on *data-driven* workflow modeling [13,20]. The capability of modeling fault handlers (e.g. automatic retry of failed tasks) is of special interest for large-scale scientific workflows [17,9].
(c) *Domain-specific graphical user interfaces*: assistance with the modeling procedure as well as the visualization of simulation results is crucial for user acceptance and success. Specially tailored techniques emerged in the field of 13C-MFA (cf. Fig. 6 for an example) [5].

The BPEL-based open-source solution *ActiveBPEL* was elected as *control-based* scientific workflow engine, allowing fine-grained modeling of large-scale 13C-MFA applications. Several valuable *ActiveBPEL* extensions are readily available, like legacy code wrapping, fault-tolerant workflow execution and on-demand Grid integration [8,9,2]. Because BPEL-defined workflows are exposed as web services in a service-oriented architecture (SOA), 13C-MFA applications can be composed of reusable sub-workflows.

1.3 Aims of this Contribution

A scientific workflow framework supports users of scientific applications by hiding technical aspects, and thus significantly reduces the complexity of applications. Even more important, processing steps become reproducible and, in turn documentable. This contribution focuses on two important aspects in the context of 13C-MFA workflows: data integration and graphical user interfaces. In Section 2, the modeling and visualization software *Omix*, the simulation toolbox *13CFLUX2*, and the scientific workflow architecture are introduced. Section 3 presents the modeling workflow as application example in detail. For the integration of existing 13C-MFA software tools into the scientific workflow framework, two aspects of implementation are focused on in detail in Section 4: first, a plug-in interface for *Omix* is presented; second, the web service integration of *13CFLUX2* into the workflow framework is demonstrated by means of the parameter fitting procedure. Finally, Section 5 concludes this contribution.

2 Ingredients for 13C-MFA

Because the 13C-MFA method is a model-based approach, adequate software tools supporting the modeling process are required. In the following, a scientific workflow framework for 13C-MFA applications is presented. This workflow system utilizes the graphical model editing and visualization software *Omix*, and the high-performance simulation toolbox *13CFLUX2*.

2.1 Model Editing and Visualization with Omix

In order to set up models of biochemical organisms in an appropriate way, expert knowledge is necessary. Therefore, modelers make heavy use of various knowledge sources, such as published literature and reaction databases. Beside the modeling process of metabolic networks, the second important task of 13C-MFA is the visualization of intermediate and final results from experimental and simulation studies. *Omix* is a highly sophisticated metabolic network editor which is developed for modeling and visualization purposes [4]. In *Omix*, results from experiment and simulation are displayed in the context of the same graphical network representation. The software is developed in Java using the Jambi wrapper of the Qt framework for the graphical user interface.

2.2 High-Performance Simulation Toolbox: 13CFLUX2

13CFLUX2 is a set of high-performance software programs implementing the 13C-MFA method [25]. *13CFLUX2* consists of about 20 distinct applications written in C++, Perl and Python. These programs are command-line applications, which can be categorized as follows:

- *Tools for Simulation and Exploration*: programs for simulation form the core of the *13CFLUX2* toolbox. Other tools in this category include Monte Carlo sampling methods for the characterization of the solution space, flexible parameterization of system equations and various methods for advanced stoichiometric analysis.
- *Tools for Parameter Fitting*: based on a model and a set of measurements, flux parameters are estimated. Different optimization strategies with specific parameterizations can be selected.
- *Tools for Statistical Analysis*: linearized statistics tools are implemented here [27], but also nonlinear Monte Carlo methods are available for estimating confidence regions of the determined flux parameters. Tools for optimal experimental design are also included [15].
- *Miscellaneous Tools*: various tools for data management, consistency checking and conversion are also part of *13CFLUX2*.

2.3 Scientific Workflows for 13C-MFA

The scientific workflow framework presented here is a recent development that aims at the adoption of scientific workflow automation to 13C-MFA applications. This framework integrates *13CFLUX2*, *Omix* and expert knowledge into scientific workflow applications.

Many software packages consist of a plenitude of programs, database and user interfaces usually developed independently. This often results in a heterogeneous software environment, making automation efforts with workflows difficult to implement. Program interfaces provided by *13CFLUX2* and *Omix* are designed for web service extensibility, thus allowing optimal integration into a distributed environment (cf. Section 4.2). Having set up a 13C-MFA SOA environment, web service-enabled programs can be orchestrated by a scientific workflow framework.

An overview of the 13C-MFA workflow architecture is depicted in Fig. 2. A database middleware provides web service access to measurement and reaction databases (top). Using this middleware interface, 13C-MFA workflows (i.e. modeling, simulation and visualization workflows) can also access public databases, e.g. KEGG. The simulation workflows (i.e. exploration, parameter fitting and statistics) are exposed as web and Grid services. With *Omix*, models for 13C-MFA simulations are graphically edited and results are visualized (middle, left to right). Scientific workflows are orchestrated using *ActiveBPEL* (bottom).

3 Metabolic Reaction Network Modeling Workflow

Having the main ingredients for 13C-MFA introduced, the core modeling workflow is selected to demonstrate workflow realization within our scientific workflow

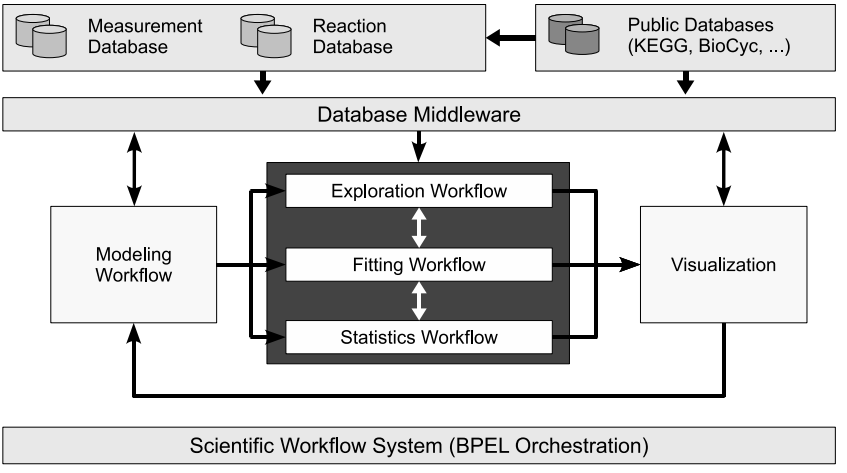

Fig. 2. The general architecture of the scientific workflow system for 13C-MFA. The building blocks are the scientific workflow system (bottom), applications (middle) and database middleware (top).

system. The modeling process basically consists of the following four steps (cf. Fig. 3):

1. **Graphical Modeling:** biochemical networks are either derived from existing models or built from scratch. This step not only handles the setup of the network topology, but also the specification of the carbon atoms' fates. Biological knowledge and various information sources are incorporated into the model, such as published data from literature or public databases.
2. **Model Configuration:** the network structure is extended by different classes of stoichiometric equality and inequality constraints. One class simply restricts the directionality of selected biochemical reactions, other classes tie together certain reaction rates or limit the range of allowed flux values.
3. **Simulation and Evaluation:** assuming that the stoichiometric constraints defined in the previous step are feasible, simulated measurements are obtained from the solution of the model equations. Combining the simulation results with measurement data obtained from isotope labeling experiments, the unknown fluxes are then estimated using an iterative, computationally intensive fitting approach. Subsequently, confidence regions of the estimated fluxes are derived using statistical methods.
4. **Visualization:** finally, simulation results are visualized. Therefore, *Omix* is employed as a specialized toolkit allowing data visualization in association with metabolic network diagrams.

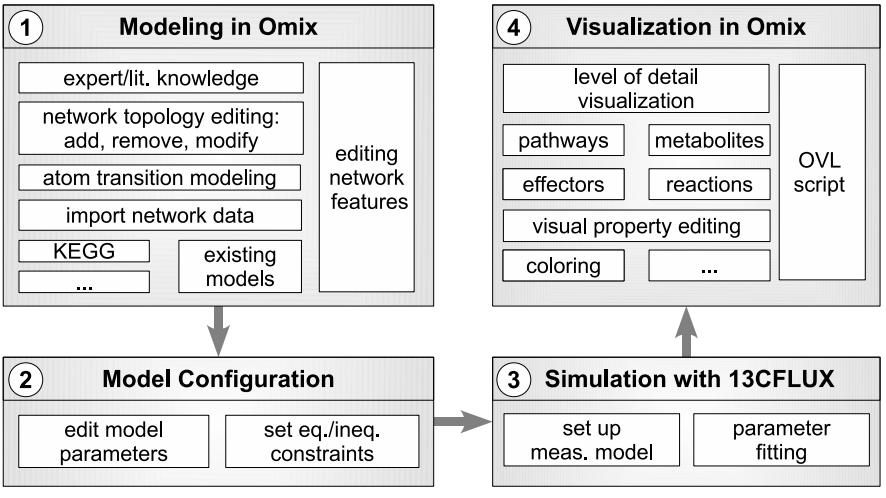

Fig. 3. A typical 13C-MFA modeling workflow. This workflow consists of the four steps model editing, configuration, parameter estimation with measurement data and visualization of simulation results.

The four steps of this workflow are discussed in detail in the following sections.

3.1 Graphical Network Modeling with Omix

An elegant way to prevent technically involved and, thus, error-prone manual editing of model description documents is the graphical editing of metabolic network models. These models are constituted by enzyme-catalyzed biochemical reactions converting substrate pools to intermediate and product pools. Key properties for the quantitative understanding of cellular metabolism are absolute metabolite *concentrations* and the reaction rates, the *fluxes*. Both, concentrations and fluxes are correlated quantities. In a graphical modeling process the modeler inserts pool and reaction symbols into a diagram and interconnects them with lines (cf. Fig. 4). The modeled network structure is easy to grasp because of its visual representation. Hence, errors in the model can be discovered very fast.

Omix mimics the functionality found in other popular vector drawing tools (see Fig. 5 a)). This warrants intuitive access to the software and eases the modeling of metabolic networks, even for users having little experience with computational tools. In principle, there are two ways to set up a network model in *Omix*:

(a) The network diagram can be drawn manually bottom up, i.e. every pool and reaction symbol is drawn individually and the connection lines are inserted one by one. Typical sources of information are network diagram taken from scientific publications or expert knowledge.
(b) Network topologies can be imported from already existing files or reaction databases and merged in whole or in part into the edited document.

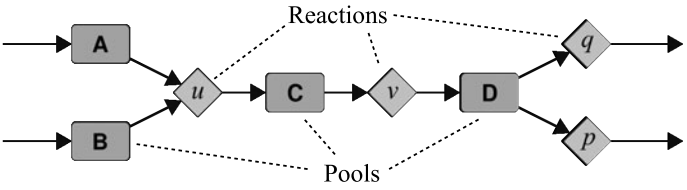

Fig. 4. Scheme of a metabolic network model consisting of metabolite pools (**A**,**B**,**C**,**D**) converted by biochemical reactions (u,v,p,q)

Both methods can be combined in any order to built up models of metabolic networks. To accelerate the drawing process, *Omix* offers several semi-automatic graph drawing techniques.

An important feature of *Omix* is that the software is equipped with a plug-in interface. The utilization of the plug-in interface facilitates the integration of *Omix* into the scientific workflow framework. For example, an *Omix* plug-in for database connectivity to KEGG[1] enables the user to inspect the metabolic networks provided by the KEGG database, to gather and select network parts for importing them into the *Omix* document. The *Omix* plug-in interface is described in more detail in Section 4.1.

3.2 Network Model Configuration

An XML-based document format called *FluxML* was developed for applications in the *13CFLUX2* environment [25]. Beside network topology information, FluxML documents contain full information required for model parameterization including stoichiometric constraints, atom transitions and measurement specifications.

In order to generate a FluxML document from the drawn network model, an *Omix* plug-in manages the import/export of FluxML documents including a comprehensive validation functionality. Additionally, a network graph is augmented with other essential information:

- *atom transitions*: carbon atom mappings of each reaction are defined in the document.
- *measurement specification*: labeling patterns for input substrates are specified. In addition, measurement values can be stored.
- *constraint and parameter configurations*: parameters constraining the network specification and flux parameter values are supplied.

The *Omix* FluxML plug-in offers dialog windows for editing all of these parameters. E.g. atom transitions can be edited graphically using the FluxML plug-in (cf. Fig. 5 b). Alternatively, parameters are adopted from an imported document. Inconsistent parameters or an invalid network topology are rejected by the validation tool included in the FluxML plug-in.

[1] http://www.genome.jp/kegg

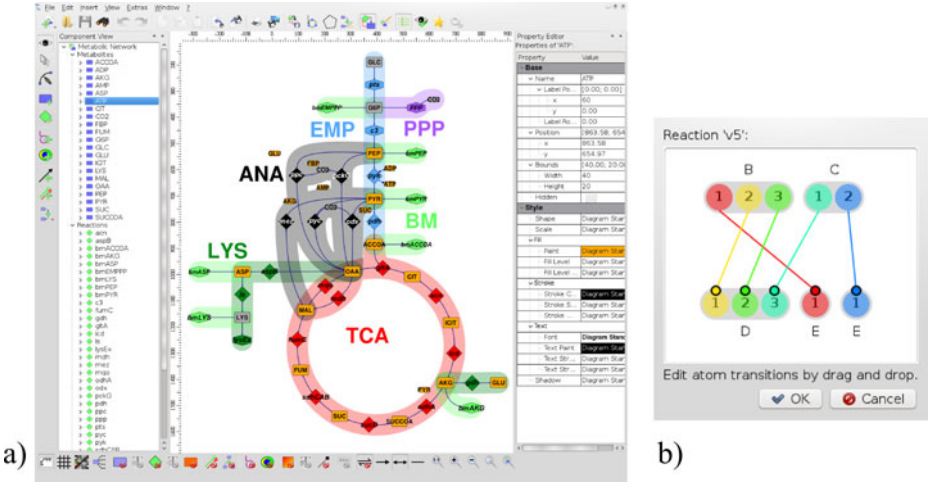

Fig. 5. Screenshot of *Omix* — a) The central window component is the drawing area for network diagrams. Toolbars containing icons around this component make various editing options available. Sidebars provide information about network properties. b) Atom transitions of the reactions can be edited graphically with *drag and drop*.

3.3 Simulation and Evaluation

Given a valid metabolic network model, fluxes can be estimated with *13CFLUX2*. For successful execution of the parameter fitting process, measurements from isotope labeling experiments need to be incorporated into the FluxML document. Simulation and parameter estimation of high-dimensional and nonlinear 13C-MFA models are computationally sophisticated procedures (cf. Fig. 1) described elsewhere [25]. Subsequent to the parameter fitting, a statistical analysis is conducted to identify the determined flux distribution's certainty. In particular, confidence intervals of all fluxes are determined.

3.4 Visualization

Simulation results are finally visualized in the *Omix* network diagram (cf. Fig. 6 for typical flux distribution maps). Visualization in *Omix* is programmable by a scripting language called *Omix Visualization Language* (OVL) and therefore fully customizable to any user requirement. In particular, OVL allows a fast and simple access to visual properties of network symbols like color, shape and line width, just to mention a few. For instance, OVL allows the user to implement interactive components, to parse data files and to assign data to visual properties of the network diagram ([4,3]).

Fig. 6 a) shows a network diagram representing the central metabolism of a microbial organism. The diagram contains metabolite pools and fluxes of central metabolic pathways connected by arrow-headed lines. If reactions are reversible,

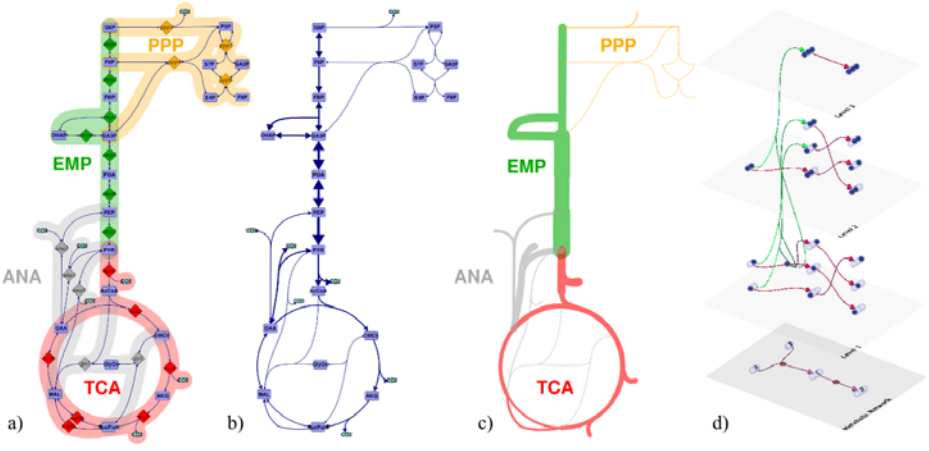

Fig. 6. Visualization in the context of MFA – a) metabolic network diagram with highlighted pathways (e.g. gluconeogenesis - green, pentose phosphate pathway - orange, citric acid cycle - red etc.); boxes show metabolites; diamond symbols represent reactions where their color codes for the pathway the reaction is assigned to. b) and c) show the network in two different levels of detail augmented with flux distribution data from a 13C-MFA. d) shows the atom transitions of a simple example network visualized in 3D by the *Omix* plug-in CumoVis.

i.e. without preferred directionality, this is indicated visually by a double arrow-headed line. For ease of recognition, biochemical reactions are often grouped to metabolic pathways. This is indicated by highlighting color strokes in the background of the pathway.

Because network diagrams can be large and very complex, *Omix* offers the option to reduce the level of detail of a network diagram by hiding whole classes of network components. In the visualization example, flux symbols as well as pathways are hidden (Fig. 6 b) and all details except pathways are concealed (Fig. 6 c), respectively.

Beside showing different levels of detail, the network is augmented with data from the simulation study. In the example, the flux value is mapped to the width of the connection lines of a reaction: line width indicates the velocity of the conversion rate (cf. Fig. 6 b). The width of the pathway stroke indicates the overall activity of reactions in the pathway (cf. Fig. 6 c). A global view of the carbon flow over the whole network is facilitated utilizing another *Omix* plug-in called CumoVis [5] (cf. Fig. 6 d).

4 Implementation Details

Having presented the workflow components for the modeling of a metabolic network, we now focus on two specific integration aspects of *Omix* and *13CFLUX2* into the scientific workflow framework.

4.1 Omix Plug-In Interface

Omix provides an interface which allows to extend the facilities of the software. This interface is called the *Omix API*. An *Omix* plug-in is a Java archive containing Java plug-in code and a specification file. The specification file describes the plug-in interface in detail, including plug-in name, license information, version and type of the plug-in. The following extension types can be realized as Omix plug-ins:

- **Interaction Extension** equip the *Omix* main window with graphical and interactive components, e.g. with menus or toolbars as document-based or document independent feature.
- **Model I/O File Filters** handle file import and export.
- **Image Export Filters** are used to implement the export of graphics into various image and animation formats, like SWF.
- **Network Communication Protocols** implement Internet transfer interfaces. For example, the network protocols SSH or SMB are available as web communication plug-ins.
- **Data Type Management** for editing plug-in-specific data types, i.e. model parameters (cf. Section 3.1).
- **Plug-in Configuration** attach custom configuration window components to the main configuration manager window of *Omix*.

An *Omix* plug-in can realize and combine an arbitrary number of these extension types. The FluxML plug-in, for instance, realizes an interaction extension and hereby provides the network validation feature in the menu bar. Simultaneously, it realizes a network I/O filter and a data type manager for the parameter modeling feature. The *KEGG Database Import* likewise implements an interaction extension using the privilege to establish database connectivity and to compose new documents in *Omix*. Because this plug-in interface is well documented any third party can develop plug-ins in order to extend Omix with specific functionality.

4.2 Web Service Implementation of the Parameter Fitting Program

As a typical *13CFLUX2* program, the parameter estimation program *fitfluxes* is chosen to exemplarily demonstrate web service extensibility. This Unix console application takes several program arguments. For basic functionality, the following arguments are required to be specified:

-i [file]: input FluxML model
-o [file]: output XML file with estimated flux parameters

Several optional command-line arguments are available, e.g. for tuning the optimization procedure:

-O [string]: select optimizer package
-g [string]: select gradient mode

A typical call of *fitfluxes* is:

```
fitfluxes -i model.fml -o out.xml -O IPOPT -g analytic
```

Here, the input (*model.fml*) and output (*out.xml*) files are specified, the optimizer package *IPOPT* [23] and the *analytic gradient* mode are used. Converting this program into a web service is accomplished by wrapping the call by a Java web service program:

```
@WebService()
public class FitFluxesWS {
  @WebMethod(operationName = "fitfluxes")
  public int fitfluxes(
    @WebParam(name = "infile") String fml,
    @WebParam(name = "outfile") String fwd,
    @WebParam(name = "optimizer") String opt,
    @WebParam(name = "gradient") String grad )
  {
    String fitfluxesCmd = ...; /* assemble the command */
    /* run fitfluxes */
    prc = Runtime.getRuntime().exec(fitfluxesCmd);
    /* handle failures, process results */
    return 0;
  }
}
```

Listing 1.1. Java web service definition for *fitfluxes*

Being an illustrative example for wrapping existing command-line programs with web services, the general procedure imposes several limitations:

- The web service is assumed to share a common networking file system. However, in a real world Grid environment, this is typically not the case. Thus, data exchange has to be modeled as well.
- A web service wrapper for each *13CFLUX2* program needs to be implemented. This manual procedure is an inflexible, laborious and, hence, error-prone procedure.
- Unix I/O streams need to be modeled as well. For example, *13CFLUX2* programs write error logging messages to the *standard error stream*.

Several solutions for wrapping legacy application code into a SOA are available, e.g. GEMLCA, Soaplab or GMS [22]. The *Legacy Code Description Language* (LCDL) is elected, because *13CFLUX2* programs can be intuitively modeled as web services in the scientific workflow framework in a graphical manner [8]. In *LCDL*, web service interfaces are designed graphically using the *Eclipse Modeling Framework*. Data transfer between *LCDL*-generated web services is realized with the *Flex-SwA* middleware [6]. With *LCDL* and *Flex-SwA*, the integration of

existing *13CFLUX2* programs into the scientific workflow architecture is easily possible.

5 Conclusions

In the contribution we sketched the implementation of the metabolic network modeling workflow within our scientific workflow framework. The modeling workflow basically consists of the following four interlinked steps: (1) visual modeling of a metabolic reaction network, (2) mathematical modeling of reactions, atom transitions and biochemical constraints, (3) evaluation of experimental data sets, i.e. the estimation of model parameters, and (4) visualization of temporary and final results. Data integration and dynamic human tasks are involved in all steps of the 13C-MFA workflow.

The presented scientific workflow framework approaches the integration of human tasks and knowledge sources by recombining well-established software tools. While leaving the core software untouched, *Omix* was extended by a flexible plug-in interface, allowing user-oriented integration of scientific knowledge sources. The integration of the high-performance simulation toolbox *13CFLUX2* into a SOA was performed by wrapping each program with a simple web service interface. Thus, these programs are seamlessly integrated into the workflow framework, allowing an intuitive usage of scientific 13C-MFA applications. The graphical web service modeling framework *LCDL* and the *Flex-SwA* data exchange middleware are used for the integration of *13CFLUX2* programs into a SOA environment. These tools have been proven to work well together in combination with the workflow orchestration software *ActiveBPEL* [8,2].

The presented modeling workflow was implemented aiming at liberating scientists from organizational and recurring tasks. Basic knowledge acquisition, routine data conversion and processing steps are handled by the scientific workflow framework. This software-aided treatment not only reduces complexity of 13C-MFA applications, but also helps to avoid manual errors. The major aim of the scientific workflow framework is automation and, in turn, acceleration of the overall processing time. At the end, this paves the way for a reproducible handling of higher-throughput data.

Future work in this area includes the implementation of a 13C-MFA database system, offering models, experimental data and simulation results. Major challenges for a 13C-MFA database middleware are user management and access control. After augmenting all *13CFLUX2* programs with web service functionality, the development of further 13C-MFA scientific workflows with automation and Grid provisioning can be easily achieved.

Acknowledgments. We would like to thank Ernst Juhnke, Tim Dörnemann and Bernd Freisleben (Dept. of Mathematics and Computer Science, University of Marburg) for their support with *LCDL*, *Flex-SwA* and *ActiveBPEL*. We thank Tobias Vehrkamp for implementing the KEGG plug-in.

References

1. Barseghian, D., Altintas, I., Jones, M.B., Crawl, D., Potter, N., Gallagher, J., Cornillon, P., Schildhauer, M., Borer, E.T., Seabloom, E.W., Hosseini, P.R.: Workflows and extensions to the kepler scientific workflow system to support environmental sensor data access and analysis. Ecological Informatics 5(1), 42–50 (2010)
2. Dörnemann, T., Juhnke, E., Freisleben, B.: On-Demand Resource Provisioning for BPEL Workflows Using Amazon's Elastic Compute Cloud. In: Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2009), pp. 140–147. IEEE Press, Los Alamitos (2009)
3. Droste, P., von Lieres, E., Wiechert, W., Nöh, K.: Customizable Visualization on Demand for Hierarchically Organized Information in Biochemical Networks. In: Barneva, R.P., Brimkov, V.E., Hauptman, H.A., Natal Jorge, R.M., Tavares, J.M.R.S. (eds.) CompIMAGE 2010. LNCS, vol. 6026, pp. 163–174. Springer, Heidelberg (2010)
4. Droste, P., Noack, S., Nöh, K., Wiechert, W.: Customizable Visualization of Multiomics Data in the Context of Biochemical Networks, pp. 21–25. IEEE Computer Society, Los Alamitos (2009)
5. Droste, P., Weitzel, M., Wiechert, W.: Visual exploration of isotope labeling networks in 3D. Bioprocess and Biosystems Engineering 31, 227–239 (2007)
6. Heinzl, S., Mathes, M., Friese, T., Smith, M., Freisleben, B.: Flex-SwA: Flexible Exchange of Binary Data Based on SOAP Messages with Attachments. In: Proc. of the IEEE International Conference on Web Services, Chicago, USA, pp. 3–10. IEEE Press, Los Alamitos (2006)
7. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic Acids Research 34, 729–732 (2006)
8. Juhnke, E., Seiler, D., Stadelmann, T., Dörnemann, T., Freisleben, B.: LCDL: An Extensible Framework for Wrapping Legacy Code. In: Proceedings of ERPAS 2009, pp. 646–650 (2009)
9. Juhnke, E., Dörnemann, T., Freisleben, B.: Fault-Tolerant BPEL Workflow Execution via Cloud-Aware Recovery Policies. In: Software Engineering And Advanced Applications, SEAA 2009, pp. 31–38 (2009)
10. Kitano, H.: Systems Biology: A Brief Overview. Science 295(5560), 1662–1664 (2002)
11. Lamprecht, A.L., Margaria, T., Steffen, B.: Bio-jETI: a framework for semantics-based service composition. BMC Bioinformatics 10(suppl. 10), S8 (2009)
12. Lamprecht, A.L., Margaria, T., Steffen, B., Sczyrba, A., Hartmeier, S., Giegerich, R.: GeneFisher-P: variations of GeneFisher as processes in Bio-jETI. BMC Bioinformatics 9(suppl. 4), S13 (2008)
13. Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y.: Scientific workflow management and the Kepler system. Concurrency and Computation: Practice and Experience 18(10), 1039–1065 (2006)
14. Mietzner, R., Karastoyanova, D., Leymann, F.: Business Grid: Combining Web Services and the Grid. In: Jensen, K., van der Aalst, W.M.P. (eds.) Transactions on Petri Nets and Other Models of Concurrency II. LNCS, vol. 5460, pp. 136–151. Springer, Heidelberg (2009)
15. Möllney, M., Wiechert, W., Kownatzki, D., de Graaf, A.A.: Bidirectional reaction steps in metabolic networks,: Optimal design of isotopomer labeling experiments. Biotechnology and Bioengineering 66(2), 86–103 (1999)

16. Neuweger, H., Albaum, S.P., Dondrup, M., Persicke, M., Watt, T., Niehaus, K., Stoye, J., Goesmann, A.: Meltdb: a software platform for the analysis and integration of metabolomics experiment data. Bioinformatics 24(23), 2726–2732 (2008)
17. Ngu, A.H., Bowers, S., Haasch, N., Mcphillips, T., Critchlow, T.: Flexible Scientific Workflow Modeling Using Frames, Templates, and Dynamic Embedding. In: Ludäscher, B., Mamoulis, N. (eds.) SSDBM 2008. LNCS, vol. 5069, pp. 566–572. Springer, Heidelberg (2008)
18. Nielsen, J., Jewett, M.C.: Impact of systems biology on metabolic engineering of Saccharomyces cerevisiae. FEMS Yeast Research 8(1), 122–131 (2008)
19. Shoshani, A., Rotem, D.: Scientific Data Management: Challenges, Technology, and Deployment. Chapman & Hall/CRC (2009)
20. Tan, W., Missier, P., Foster, I., Madduri, R., De Roure, D., Goble, C.: A Comparison of Using Taverna and BPEL in Building Scientific Workflows: the case of caGrid. Concurr. Comput.: Pract. Exper. 22(9), 1098–1117 (2010)
21. Tang, Y.J., Martin, H.G., Myers, S., Rodriguez, S., Baidoo, E.E.K., Keasling, J.D.: Advances in analysis of microbial metabolic fluxes via ^{13}C isotopic labeling. Mass Spectrometry Reviews 28(2), 362–375 (2009)
22. Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M.: Workflows for e-Science: Scientific Workflows for Grids. Springer, New York (2006)
23. Wächter, A.: An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering. Ph.D. thesis, Carnegie Mellon University (2002)
24. Wang, J., Crawl, D., Altintas, I.: Kepler + Hadoop: A General Architecture Facilitating Data-Intensive Applications in Scientific Workflow Systems. In: WORKS 2009: Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science, pp. 1–8. ACM, New York (2009)
25. Weitzel, M.: High Performance Algorithms for Metabolic Flux Analysis. Ph.D. thesis, University of Siegen, Germany (2009)
26. Wiechert, W.: ^{13}C Metabolic Flux Analysis. Metababolic Engineering 3(3), 195–206 (2001)
27. Wiechert, W., Siefke, C., de Graaf, A.A., Marx, A.: Bidirectional reaction steps in metabolic networks, Part II: Flux estimation and statistical analysis. Biotechnology and Bioengineering 55(1), 118–135 (1997)
28. Zamboni, N., Fendt, S.M., Rühl, M., Sauer, U.: ^{13}C-based metabolic flux analysis. Nature Protocols 4(6), 878–892 (2009)