Scientific Workflows and XMDD

Anna-Lena Lamprecht and Tiziana Margaria

Chair for Service and Software Engineering, Potsdam University,
Potsdam, D-14482, Germany
{lamprecht,margaria}@cs.uni-potsdam.de

Abstract. A major part of the scientific experiments that are carried out today requires thorough computational support. While database and algorithm providers face the problem of bundling resources to create and sustain powerful computation nodes, the users have to deal with combining sets of (remote) services into specific data analysis and transformation processes. Today's attention to "big data" amplifies the issues of size, heterogeneity, and process-level diversity/integration. In the last decade, especially workflow-based approaches to deal with these processes have enjoyed great popularity. This book concerns a particularly agile and model-driven approach to manage scientific workflows that is based on the XMDD paradigm. In this chapter we explain the scope and purpose of the book, briefly describe the concepts and technologies of the XMDD paradigm, explain the principal differences to related approaches, and outline the structure of the book.

Keywords: scientific workflows, workflow management systems, model-driven design, XMDD.

1 Introduction

Scientific algorithms, tools, and databases have assisted researchers of all disciplines in different phases of their data analyses for over two decades now. These tools and databases are usually not installed locally, because handling the large amounts of data, possibly together with high time and space complexities of the underlying algorithms, requires the software to be run on supercomputers or appropriate clusters. Grid computing, the precursor of the cloud for collective scientific computing facilities, has been an outcome of this need. Tool providers offer interfaces to execute their service programs remotely, often as web interfaces or web services.

Working with a large set of different services needed for an analysis process and coordinating the use of computational tools and information repositories distributed throughout the world is a cumbersome and error-prone task when the individual steps are carried out manually. Challenging data analysis tasks typically require complex processes and specific care in combining, monitoring, and documenting the single analysis steps. Therefore, frameworks that provide the means for automating complex analysis processes involving a number of

A.-L. Lamprecht et al. (Eds.): Process Design for Natural Scientists, CCIS 500, pp. 1–13, 2014.
DOI: 10.1007/978-3-662-45006-2_1

heterogeneous services enjoy great popularity. In particular, workflow modeling techniques of all flavors are increasingly successful. Essentially, they aim at simplifying access to the services, so that scientists with little programming experience, but with well-founded knowledge about the domain and the specific process, can orchestrate cohorts of services easily in order to achieve their goals.

Services and service collections providing access to remote data repositories or to computational tools form the basic components from which workflows are built. Workflow management systems then provide composition interfaces where users can select and combine services into workflows. The assembly often happens graphically by placing service "proxies" as building blocks on a canvas, and connecting them either according to the flow of control or according to the flow of data [1]. Usually these systems also provide the means for direct execution of the workflow model, and several more features that add convenience for the users in different ways.

A principal question regarding workflow management systems concerns their scope of applicability: In which application areas can they be used most fruitfully, in comparison to direct programming or scripting? The ROI of workflow modeling systems depends on the levels of complexity, reuse, quality, assurance and more. Certainly, data analyses or analysis steps that are completely fixed and can be repeated or reused without any change could rather be implemented conventionally, and then used as individual software programs or reused as atomic services within larger workflows. However, whenever there exist (at least potential) variants of a computational experiment, the workflow approach is beneficial due to its agile, modular assembly and composition style. In fact, understanding how much this facilitates an easier understanding and change of the analysis processes also by users that are not specially trained in programming was one of the major motivations for the study underlying this book.

This book introduces its readers to the field of *model-driven design and management of scientific workflows* in a case study-based fashion. It collects the outcomes of the practical project work carried out within the one-semester course "Process modeling in natural sciences". We started teaching this course in 2007, predominantly to Master students of the "Bioinformatics" and "Geoinformation and Geovisualization" programs at the University of Potsdam, gradually transforming into a project-oriented class that applies a "learn by doing approach". The course addresses processes and workflow management concepts and applies them in the project reality, acknowledging the fact that the notions of service-orientation and workflow modeling are increasingly relevant and widely adopted in the scientific community. Still, to our knowledge this approach to teaching informatics-heavy process science to experimental scientists in an individual and hands-on applied fashion is so far unique.

In This Book

The heart of the process part of the course is the practical project work carried out by the students. They design and implement workflows for self-defined applications stemming from their respective areas of expertise. A collection of

projects developed in the summer term 2012 forms the basis for this book. This collection of scientific workflow applications provides insights and experience about the modeling of scientific workflows from the users' point of view as well as from the computer-science perspective.

Besides documenting the impact high-level that workflow modeling might have on the work of natural scientists, this book serves three major purposes:

1. It acts as a **primer for practitioners** who are interested to learn how to think in terms of services and workflows when facing domain-specific scientific processes.
2. It provides interesting **material for readers** already familiar with this kind of tools, because it introduces systematically both the technologies used in each case study and the basic concepts behind them.
3. As the addressed thematic field becomes increasingly relevant for lectures in both computer science and experimental sciences, it also provides helpful **material for teachers** that plan similar courses.

To the best of our knowledge, no other book currently available serves these purposes.

In "Workflows for e-Science – Scientific Workflows for Grids" [2], Taylor et al. present a comprehensive and heterogeneous collection of applications from different scientific disciplines that use a variety of different methods and technologies. Considering our book as a primer, this is a nice complement providing a wider survey of the field to readers already familiar with workflow technologies and process- and service-oriented thinking.

In "Scientific Workflows - Programming, Optimization, and Synthesis with ASKALON and AWDL" [3], Qin and Fahringer provide a well-elaborated and comprehensive introduction to basic and advanced topics of scientific workflow management in general and to the ASKALON framework and its extensions in particular. For practitioners not trained in computer science and new to the field of workflows, it may again be a nice second step, being more technical and "heavyweight" and providing a very limited number of concrete examples aimed at illustrating specific technical capabilities of the specific framework.

This book is also different from the available literature on graphical programming systems like, e.g., LabView [1]. In these systems, workflows are used as visual program representations, but they still require a deep technical knowledge of the programming issues, without actually lifting the level of abstraction in handling the technology to render it accessible by users that are not specially trained in IT or programming. Enabling the non-IT experts to proficiently work with workflows without undergoing a prior deep technical training is, on the contrary, the principal focus of the service-oriented methods that we apply in our workflow modeling concepts and that we demonstrate in this collection of case studies.

Our book focuses on what is needed to be a primer to the field of scientific workflows. We start with a systematic but technically simplified introduction to our methods and technologies for process management and workflow design.

[1] `http://www.ni.com/labview/`

They are then consistently applied by all the case studies. All the applications are presented in a uniformly structured fashion in order to facilitate the reader's understanding and conceptualization. As such, the book can be used as a text book for similar courses for students of experimental sciences, but it does also provide helpful additional material for anybody dealing with workflow modeling for other reasons.

In the following we introduce our specific approach to scientific workflow modeling (Section 2) and some selected applications (Section 3), give a brief survey of related approaches (Section 4) and outline the structure of the book (Section 5).

2 Extreme Model-Driven Design of Scientific Workflows

XMDD (eXtreme Model-Driven Design), introduced in [4] and refined in [5–9], is a methodology aimed at the easy uptake by practitioners that wish a fast but high-quality and high-assurance service assembly. Simplicity [7] is here the key guiding principle. According to the motto "*easy for the many, difficult for the few*", XMDD is an approach that systematically reinforces the intuition of the users when dealing with a graphical workflow design framework that eases service composition, and at the same time it provides the high-assurance level of quality that is usually based on formal description languages and formal methods, that are however normally inaccessible to non-IT experts, and challenging even for the average programmer and software engineer. The simplicity tradeoff we adopt foresees the use of *embedded* formal descriptions and formal methods (the "difficult" part), whereby due to the embedding, users do not need to be aware of these traits and capabilities that are for them kept under the hood of the tool they use, thereby achieving a simplicity effect.

Extreme Model Driven Design (XMDD) leverages this approach of separation of concerns by combining the decisive traits of several modern software engineering schools into a coherent paradigm that efficiently and effectively leverages their characteristic contributions. XMDD builds on ideas taken from

- eXtreme programming [10], for providing immediate feedback through requirement and design validation by means of model tracing, simulation and early testing, virtualizing the implementation of functionality,
- service orientation [11], for virtualizing the implementation of functionality,
- aspect orientation [12], for treating crosscutting as well as role specific concerns modularly, and
- model-driven design [13], for controlling the overall development at the modeling level.

For the process definition and management according to the XMDD paradigm we use a multi-purpose domain-independent modeling framework: the Java Application Building Center (jABC) [14]. Thank to a rich set of associated embedded technologies and plugins, it provides a comprehensive yet intuitive graphical framework that users use to integrate services, build workflows (in the form of

directed graphs) from the components that result from the integration, analyze and execute them, and finally deploy and provision them in the form of applications or services. It is based on well-established software technology, as described above, and has been used successfully in different application domains.

The aim of this software framework is to support in their daily work and boost the productivity of application experts without specific IT knowledge. Such users use the jABC environment to model and design workflows in a service-oriented fashion and the jETI service integration technology [15, 16] to execute remote tools and services as if they were locally available. These are the two central technologies used in this book.

3 Our Experience So Far

With jABC and jETI we participated in different conferences and initiatives in the area of scientific workflows, such as the NETTAB 2007 workshop session "From Components to Processes in Bioinformatics" [17], the SWAT4LS Workshops [18–20] and the 2010 Biohackathon [21]. The exchange of experiences with other researchers from the scientific workflow community and the work on several case studies during the last years has taught us how to best proceed to service-enable different subdomains of scientific processes and workflows and how to use our technology to orchestrate complex analyses of experimental data. Some of our previous case studies are particularly useful for a comparison with the cases presented in this book as well as for in-depth further study:

- Our first case study in the bioinformatics application domain concerned a distributed workflow for validation of orthologous gene structures among a selection of higher organisms [22]. This study demonstrates how to model hierarchical bioinformatics workflows in the jABC framework, based on external tools and resources in the same fashion as described in [23].
- The study described in [24] deals with the preprocessing and statistical analysis of liquid chromatography/mass spectrometry (LC/MS) data using the statistics language GNU R [25] and in particular the XCMS package [26]. In this application, the required R functionality is integrated using the jETI technology, while the jABC again serves as workflow definition environment.
- GeneFisher-P [27] is a service-oriented and workflow-based re-implementation of the GeneFisher [28, 29] web application for PCR primer design, realized using jABC and jETI. It makes it possible to run the primer design process in a batch processing manner, and thus to design primers for large amounts of input sequences automatically. Furthermore, it facilitates the user-level definition of workflow variants, useful for customizing workflows by including alternative services for the individual steps of the design process. This has been done, for instance, by the biologist Janus Borner in his Diploma thesis at the university of Hamburg [30]: he integrated specific scripts for the backtranslation steps and used them to build his specific variants of the original workflow.

- Woven around workflows for multiple sequence alignment - a commonly known computation that is in fact part of many analyses in genomics, proteomics and transcriptomics - the study described in [31] focuses on illustrating how the framework enables end users that are not IT experts to define, analyze, execute, modify, and interactively develop bioinformatics analysis processes. These workflows are at the same time particularly suited to demonstrate the flexibility and agility that analyses gain in the framework, especially in contrast to the completely predefined service programs that are widely spread in the scientific domain.
- Flux-P [32] is an approach to automate and standardize ^{13}C-based metabolic flux analysis [33] using jABC and jETI. Flux-P is currently exemplarily based on the FiatFlux software [34]. It demonstrates how to create services that carry out the different analysis steps autonomously and how to subsequently assemble them into software workflows that perform automated, high-throughput intracellular flux analysis of high quality and reproducibility. Besides significant acceleration and standardization of the data analysis, the agile workflow-based realization supports flexible changes of the analysis workflows on the user level, making it easy to perform custom analyses.
- The Climate Impacts: Global and Regional Adaptation Support Platform (ci:grasp) [35][2] is a web-based climate information service for exploring climate change related information in its geographical context. The project described in [36] uses the jABC workflow modeling and execution framework to make flexibilized versions of the processes implemented in ci:grasp available to the scientific community, enabling users to flexibly define and adapt the workflows according to their specific needs.

An essential characteristic of scientific workflows, including the case studies listed above, is their being subject to frequent changes. This demands for workflow systems that enable a flexible workflow development style, where exchanging services and building variations of workflows is easy for the user. Our experiences with students and project partners using jABC and jETI provide evidence that an accessible level of abstraction has indeed been reached, and that it enables an agile handling of workflows. For instance, the GeneFisher-P and FiatFlux-P workflows were frequently adapted by their users (biology diploma and bio-engineering PhD students without specific computer science education) according to changing experimental setups. What is more, similar workflows were built by (computer science and other) students autonomously in the scope of different lectures and projects.

While the majority of the the applications listed above stem from the bioinformatics domain (note there is also a domain-specific incarnation of the normally domain-independent jABC and jETI frameworks in the bioinformatics area that is called Bio-jETI [37]), this book comprises a greater share of workflows from the geovisualization and image processing domains. This demonstrates that the framework is easily applicable also in other scientific domains.

[2] `http://www.cigrasp.org`

4 Relationship to Other Scientific Workflow Systems

After almost two decades of research and development in the field of scientific workflows, several workflow design and management environments with different characteristics are available for scientific applications. This section briefly surveys some other existing approaches to scientific workflow management and compares them with regard to some aspects that are important in the scope of this book with the XMDD-based approach followed in the jABC framework. More elaborate comparisons and discussions are available, for example, in [1, 38].

Kepler [39], Taverna [40–43], Triana [44] and Pegasus [45], for example, are popular scientific workflow systems born on top of distributed computing projects (such as Grids and Clusters). They offer successful workflow design environments and workflow enactment engines. Kepler, for example, is internally based on Ptolemy II [46], an actor-based environment for embedded system design, and its native actors concern basic data management operations on a grid. Accordingly, its workflow definition component is targeted towards a grid management level, which is finer granular and more technical than the services that jABC and jETI address. In fact, which jABC/jETI we explicitly strive for an adequate end-user-level granularity, and for maintaining the full power of general-purpose model-driven design when adapting the framework to a particular scientific domain. This enables a very high level of agility in the workflow design process, which is still an uncommon characteristic for the state-of-the-art scientific workflow management tools.

In contrast to the jABC, the provenance of most scientific workflow frameworks is neither from a software engineering/programming environment background, nor from a process model semantics or formal verification culture. The traditional Scientific Data Management community from which these tools arose is in fact historically data-oriented, and more interested in data distribution issues than on clean semantics of the behavioral aspects of a workflow.

The frameworks listed above are in fact inherently data flow-oriented workflow systems. This is indeed the most substantial difference to our framework, where the models define the control flow of a workflow. Both approaches are usually considered to be capable of expressing the same processes. In practice, however, when using the data-flow approach there are limitations with respect to the inclusion of elaborate control structures (cf., e.g. [38]). In the jABC a set of different common control structures is already available and shared among its different application domains. They make it easy to model sophisticated processes, for instance with different execution traces depending on the kind of input data, or with iterations or recursions over sets of data. The data itself is managed within an execution context of the model, and uses identifiers similar to variables to refer to particular data items. In our experience, the control flow-oriented environment for service design and analysis is an evolution step as scientific workflows become increasingly networked, parallel, conditional, event-driven, recursive, and asynchronous. This is the kind of complexity sources whose control is at the core of the jABC's strengths.

Additionally to the agile XMDD-based workflow development approach described and illustrated in this book, our framework comprises many more features and capabilities that deal with the application of (especially constraint-based) formal methods to support the design of scientific workflows [38, 47–49]. In fact, the jABC has been built with a focus on formal verification capability (cf. [50–52]). A clear formal semantics of the models and of the notion of composition provide the basis for the formal analysis and verification of properties of the designed workflows based on (automatic) mathematical proofs. To the best of our knowledge, the additional benefits offered by the jABC plugins for verification, synthesis and code generation are so far still unique among scientific workflow systems. In their intention they are related to approaches for semantics-based service discovery like in BioMoby [53–55], or for automatic workflow composition like in jORCA [56, 57], SADI/SHARE [55, 58, 59], Wings [60] or ASKALON [3]. Comprehensive case studies on these topics are currently being carried out, bound to become the subject of a separate volume.

5 Outline of the Book

This book combines an introduction to service- and process-oriented thinking for scientific workflows, of the corresponding technology, and a gallery of applications that have been developed using these means. Thus, it can be used as a collection of examples and case studies of scientific workflows, and also serve as an introduction to the technical aspects of domain-specific modeling with the jABC framework.

The book is organized in three major parts:

- Part I (Framework, comprising [61–64]) introduces the methodologies and technologies that we use. It also clarifies the concrete setup available during the course and discusses lessons learned and future perspectives.
- Part II (Bioinformatics Applications, comprising [65–70]) contains the reports on the 6 student projects concerning applications from the bioinformatics domain.
- Part III (Geovisualization Applications, comprising [71–79]) is composed from the reports on the 9 applications from the geoinformation and visualization domain.

Each of the articles in Part II and III is supplemented with a small "identikit" box, which is placed between the conclusion section and the references. In order to make the individual article accessible for readers who do not work through the whole book, it briefly summarizes the context of the study, gives a basic profile of the described project and points to further related work.

References

1. Lamprecht, A.L., Margaria, T., Steffen, B.: Bioinformatics: Processes and Workflows. In: Laplante, P.A. (ed.) Encyclopedia of Software Engineering, pp. 118–130. Taylor & Francis (November 2010)

2. Taylor, I.J., Deelman, E., Gannon, D.B., Shields, M.: Workflows for E-Science: Scientific Workflows for Grids. Springer (2007)
3. Qin, J., Fahringer, T.: Scientific Workflows - Programming, Optimization, and Synthesis with ASKALON and AWDL. Springer, Heidelberg (2012)
4. Margaria, T., Steffen, B.: Agile IT: Thinking in User-Centric Models. In: Margaria, T., Steffen, B. (eds.) Leveraging Applications of Formal Methods, Verification and Validation. CCIS, vol. 17, pp. 490–502. Springer, Heidelberg (2009)
5. Margaria, T., Steffen, B.: Business Process Modelling in the jABC: The One-Thing-Approach. In: Cardoso, J., van der Aalst, W. (eds.) Handbook of Research on Business Process Modeling. IGI Global (2009)
6. Margaria, T., Steffen, B.: Continuous Model-Driven Engineering. IEEE Computer 42(10), 106–109 (2009)
7. Margaria, T., Steffen, B.: Simplicity as a Driver for Agile Innovation. Computer 43(6), 90–92 (2010)
8. Steffen, B., Margaria, T., Wagner, C.: 94. In: Round-Trip Engineering, pp. 1044–1055. Taylor & Francis (2010)
9. Margaria, T., Steffen, B.: Service-Orientation: Conquering Complexity with XMDD. In: Hinchey, M., Coyle, L. (eds.) Conquering Complexity, pp. 217–236. Springer, London (2012)
10. Beck, K., Andres, C.: Extreme programming explained: embrace change. Addison-Wesley Professional (2004)
11. Margaria, T., Steffen, B., Reitenspiess, M.: Service-Oriented Design: The Roots. In: Benatallah, B., Casati, F., Traverso, P. (eds.) ICSOC 2005. LNCS, vol. 3826, pp. 450–464. Springer, Heidelberg (2005)
12. Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J.M., Irwin, J.: Aspect-oriented programming. In: Akit, M., Matsuoka, S. (eds.) ECOOP 1997. LNCS, vol. 1241, pp. 220–242. Springer, Heidelberg (1997)
13. Schmidt, D.C.: Guest Editor's Introduction: Model-Driven Engineering. IEEE Computer 39(2), 25–31 (2006)
14. Steffen, B., Margaria, T., Nagel, R., Jörges, S., Kubczak, C.: Model-Driven Development with the jABC. In: Bin, E., Ziv, A., Ur, S. (eds.) HVC 2006. LNCS, vol. 4383, pp. 92–108. Springer, Heidelberg (2007)
15. Steffen, B., Margaria, T., Braun, V.: The Electronic Tool Integration platform: concepts and design. International Journal on Software Tools for Technology Transfer (STTT) 1(1-2), 9–30 (1997)
16. Margaria, T., Nagel, R., Steffen, B.: jETI: A Tool for Remote Tool Integration. In: Halbwachs, N., Zuck, L.D. (eds.) TACAS 2005. LNCS, vol. 3440, pp. 557–562. Springer, Heidelberg (2005)
17. Romano, P., Schrder, M., Cannata, N., Signore, O. (eds.): Proceedings of the 7th International Workshop NETTAB 2007: A Semantic Web for Bioinformatics - Goals, Tools, Systems, Applications (June 2007)
18. Burger, A., Paschke, A., Romano, P., Splendiani, A.: Semantic Web Applications and Tools for Life Sciences 2008. In: Proc. of 1st Workshop SWAT4LS 2008. CEUR Workshop Proceedings, Edinburgh (November 2008)
19. Marshall, M.S., Burger, A., Romano, P., Paschke, A., Splendiani, A.: Semantic Web Applications and Tools for Life Sciences 2009. In: Proc. of 2nd Workshop SWAT4LS 2009. CEUR Workshop Proceedings, Amsterdam, The Netherlands (November 2009)
20. Burger, A., Marshall, M., Romano, P., Paschke, A.: Proceedings of 3rd Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2010). In: CEUR Workshop Proceedings, Berlin, Germany (December 2010)

21. Katayama, T., Wilkinson, M., Micklem, G., Kawashima, S., Yamaguchi, A., Nakao, M., Yamamoto, Y., Okamoto, S., Oouchida, K., Chun, H.W., Aerts, J., Afzal, H., Antezana, E., Arakawa, K., Aranda, B., Belleau, F., Bolleman, J., Bonnal, R., Chapman, B., Cock, P., Eriksson, T., Gordon, P., Goto, N., Hayashi, K., Horn, H., Ishiwata, R., Kaminuma, E., Kasprzyk, A., Kawaji, H., Kido, N., Kim, Y., Kinjo, A., Konishi, F., Kwon, K.H., Labarga, A., Lamprecht, A.L., Lin, Y., Lindenbaum, P., McCarthy, L., Morita, H., Murakami, K., Nagao, K., Nishida, K., Nishimura, K., Nishizawa, T., Ogishima, S., Ono, K., Oshita, K., Park, K.J., Prins, P., Saito, T., Samwald, M., Satagopam, V., Shigemoto, Y., Smith, R., Splendiani, A., Sugawara, H., Taylor, J., Vos, R., Withers, D., Yamasaki, C., Zmasek, C., Kawamoto, S., Okubo, K., Asai, K., Takagi, T.: Lamprecht: The 3rd DBCLS BioHackathon: improving life science data integration with Semantic Web technologies. Journal of Biomedical Semantics 4(1) (2013)
22. Margaria, T., Kubczak, C., Njoku, M., Steffen, B.: Model-based Design of Distributed Collaborative Bioinformatics Processes in the jABC. In: Proceedings of the 11th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2006), pp. 169–176. IEEE Computer Society, Los Alamitos (August 2006)
23. Steffen, B., Margaria, T., Braun, V., Kalt, N.: Hierarchical Service Definition. Annual Review of Communications of the ACM 51, 847–856 (1997)
24. Kubczak, C., Margaria, T., Fritsch, A., Steffen, B.: Biological LC/MS Preprocessing and Analysis with jABC, jETI and xcms. In: Proceedings of the 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006), pp. 308–313. IEEE Computer Society, Paphos (2006)
25. The R Project for Statistical Computing, `http://www.r-project.org/` (Online; last accessed May 5, 2013)
26. Smith, C.A., Want, E.J., O'Maille, G., Abagyan, R., Siuzdak, G.: XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification. Analytical Chemistry 78(3), 779–787 (2006); PMID: 16448051
27. 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)
28. Giegerich, R., Meyer, F., Schleiermacher, C.: GeneFisher – software support for the detection of postulated genes. In: Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB), vol. 4, pp. 68–77 (1996)
29. Hagemeier, D.: GeneFisher2 - an AJAX based implementation of GeneFisher-P. Bachelor's thesis, University Bielefeld, Faculty of Technology (December 2006)
30. Borner, J.: A molecular approach to chelicerate phylogeny. Diploma thesis, Universität Hamburg (2010)
31. Lamprecht, A.-L., Margaria, T., Steffen, B.: Seven Variations of an Alignment Workflow - An Illustration of Agile Process Design and Management in Bio-jETI. In: Măndoiu, I., Wang, S.-L., Zelikovsky, A. (eds.) ISBRA 2008. LNCS (LNBI), vol. 4983, pp. 445–456. Springer, Heidelberg (2008)
32. Ebert, B.E., Lamprecht, A.L., Steffen, B., Blank, L.M.: Flux-P: Automating Metabolic Flux Analysis. Metabolites 2(4), 872–890 (2012)
33. Wiechert, W.: 13C metabolic flux analysis. Metabolic Engineering 3(3), 195–206 (2001)
34. Zamboni, N., Fischer, E., Sauer, U.: FiatFlux a software for metabolic flux analysis from 13C-glucose experiments. BMC Bioinformatics 6, 209 (2005)

35. Wrobel, M., Bisaro, A., Reusser, D., Kropp, J.P.: Novel Approaches for Web-Based Access to Climate Change Adaptation Information MEDIATION Adaptation Platform and ci:grasp-2. In: Hřebíček, J., Schimak, G., Kubásek, M., Rizzoli, A.E. (eds.) ISESS 2013. IFIP AICT, vol. 413, pp. 489–499. Springer, Heidelberg (2013)
36. Al-areqi, S., Kriewald, S., Lamprecht, A.L., Reusser, D., Wrobel, M., Margaria, T.: Agile Workflows for Climate Impact Risk Assessment based on the ci:grasp Platform and the jABC Modeling Framework. In: International Environmental Modelling and Software Society (iEMSs) 7th Intl. Congress on Env. Modelling and Software (accepted, 2014)
37. Margaria, T., Kubczak, C., Steffen, B.: Bio-jETI: A service integration, design, and provisioning platform for orchestrated bioinformatics processes. BMC Bioinformatics 9(suppl. 4), S12 (2008)
38. Lamprecht, A.-L.: User-Level Workflow Design. LNCS, vol. 8311. Springer, Heidelberg (2013)
39. Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludscher, B., Mock, S.: Kepler: An Extensible System for Design and Execution of Scientific Workflows. In: Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM 2004), pp. 21–23. IEEE Computer Society (June 2004)
40. Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., Li, P.: Taverna: A tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20(17), 3045–3054 (2004)
41. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li, P., Oinn, T.: Taverna: A tool for building and running workflows of services. Nucleic Acids Research 34(Web Server), W729–W732 (2006)
42. Missier, P., Soiland-Reyes, S., Owen, S., Tan, W., Nenadic, A., Dunlop, I., Williams, A., Oinn, T., Goble, C.: Taverna, reloaded. In: Gertz, M., Ludäscher, B. (eds.) SSDBM 2010. LNCS, vol. 6187, pp. 471–481. Springer, Heidelberg (2010)
43. Wolstencroft, K., Haines, R., Fellows, D., Williams, A., Withers, D., Owen, S., Soiland-Reyes, S., Dunlop, I., Nenadic, A., Fisher, P., Bhagat, J., Belhajjame, K., Bacall, F., Hardisty, A., Nieva de la Hidalga, A., Balcazar Vargas, M.P., Sufi, S., Goble, C.: The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. Nucleic Acids Research 41(W1), W557–W561 (2013)
44. Taylor, I., Shields, M., Wang, I., Harrison, A.: The Triana Workflow Environment: Architecture and Applications. In: Workflows for e-Science, pp. 320–339. Springer, New York (2007)
45. Deelman, E., Singh, G., Hui Su, M., Blythe, J., Gil, A., Kesselman, C., Mehta, G., Vahi, K., Berriman, G.B., Good, J., Laity, A., Jacob, J.C., Katz, D.S.: Pegasus: A framework for mapping complex scientific workflows onto distributed systems. Scientific Programming Journal 13, 219–237 (2005)
46. Eker, J., Janneck, J., Lee, E., Liu, J., Liu, X., Ludvig, J., Neuendorffer, S., Sachs, S., Xiong, Y.: Taming heterogeneity - the Ptolemy approach. Proceedings of the IEEE 91(1), 127–144 (2003)
47. Lamprecht, A.L., Margaria, T., Steffen, B.: Bio-jETI: A framework for semantics-based service composition. BMC Bioinformatics 10(suppl. 10), 8 (2009)
48. Lamprecht, A.L., Naujokat, S., Margaria, T., Steffen, B.: Semantics-based composition of EMBOSS services. Journal of Biomedical Semantics 2(suppl. 1), S5 (2011)

49. Lamprecht, A.L., Naujokat, S., Steffen, B., Margaria, T.: Constraint-Guided Workflow Composition Based on the EDAM Ontology. In: Burger, A., Marshall, M.S., Romano, P., Paschke, A., Splendiani, A. (eds.) Proceedings of the 3rd International Workshop on Semantic Web Applications and Tools for Life Sciences (SWAT4LS 2010), vol. 698. CEUR Workshop Proceedings (December 2010)
50. Steffen, B., Margaria, T., Claßen, A., Braun, V.: Incremental Formalization: A Key to Industrial Success. Software - Concepts and Tools 17(2), 78–95 (1996)
51. Jonsson, B., Margaria, T., Naeser, G., Nyström, J., Steffen, B.: Incremental requirement specification for evolving systems. Nordic J. of Computing 8, 65–87 (2001)
52. Steffen, B., Margaria, T., von der Beeck, M.: Automatic synthesis of linear process models from temporal constraints: An incremental approach. In: ACM/SIGPLAN International Workshop on Automated Analysis of Software, AAS 1997 (1997)
53. Wilkinson, M.D., Links, M.: BioMOBY: An open source biological web services proposal. Briefings in Bioinformatics 3(4), 331–341 (2002)
54. DiBernardo, M., Pottinger, R., Wilkinson, M.: Semi-automatic web service composition for the life sciences using the BioMoby semantic web framework. Journal of Biomedical Informatics 41(5), 837–847 (2008)
55. Withers, D., Kawas, E., McCarthy, L., Vandervalk, B., Wilkinson, M.: Semantically-guided workflow construction in Taverna: the SADI and BioMoby plug-ins. In: Margaria, T., Steffen, B. (eds.) ISoLA 2010, Part I. LNCS, vol. 6415, pp. 301–312. Springer, Heidelberg (2010)
56. Karlsson, J., Martín-Requena, V., Ríos, J., Trelles, O.: Workflow composition and enactment using jORCA. In: Margaria, T., Steffen, B. (eds.) ISoLA 2010, Part I. LNCS, vol. 6415, pp. 328–339. Springer, Heidelberg (2010)
57. Martín-Requena, V., Ríos, J., García, M., Ramírez, S., Trelles, O.: jORCA: easily integrating bioinformatics Web Services. Bioinformatics 26(4), 553–559 (2010)
58. Wilkinson, M.D., Vandervalk, B., McCarthy, L.: SADI Semantic Web Services - 'cause you can't always GET what you want! In: Proceedings of the IEEE Services Computing Conference, APSCC 2009, December 7-11, pp. 13–18. IEEE Asia-Pacific, Singapore (2009)
59. Wilkinson, M.D., Vandervalk, B., McCarthy, L.: The Semantic Automated Discovery and Integration (SADI) Web service Design-Pattern, API and Reference Implementation. Journal of Biomedical Semantics 2(1), 8 (2011)
60. Gil, Y., Ratnakar, V., Deelman, E., Mehta, G., Kim, J.: Wings for Pegasus: creating large-scale scientific applications using semantic representations of computational workflows. In: Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence, vol. 2, pp. 1767–1774. AAAI Press (2007)
61. Lamprecht, A.L., Margaria, T.: Scientific Workflows and XMDD. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 1–13. Springer, Heidelberg (2014)
62. Lamprecht, A.L., Margaria, T., Steffen, B.: Modeling and Execution of Scientific Workflows in the jABC Framework. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 14–29. Springer, Heidelberg (2014)
63. Lamprecht, A.L., Wickert, A.: The Course's SIB Libraries. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 30–44. Springer, Heidelberg (2014)
64. Lamprecht, A.L., Wickert, A., Margaria, T.: Lessons Learned. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 45–64. Springer, Heidelberg (2014)

65. Reso, J.: Protein Classification Workflow. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 65–72. Springer, Heidelberg (2014)
66. Blaese, L.: Data Mining for Unidentified Protein Sequences. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 73–87. Springer, Heidelberg (2014)
67. Schulze, G.: Workflow for Rapid Metagenome Analysis. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 88–100. Springer, Heidelberg (2014)
68. Lis, M.: Constructing a Phylogenetic Tree. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 101–109. Springer, Heidelberg (2014)
69. Vierheller, J.: Exploratory Data Analysis. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 110–126. Springer, Heidelberg (2014)
70. Schtt, C.: Identification of Differentially Expressed Genes. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 127–139. Springer, Heidelberg (2014)
71. Kuntzsch, C.: Visualization of Data Transfer Paths. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 140–148. Springer, Heidelberg (2014)
72. Hibbe, M.: Spotlocator Project Documentation. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 149–158. Springer, Heidelberg (2014)
73. Teske, D.: Geocoder Accuracy Ranking. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 159–170. Springer, Heidelberg (2014)
74. Sens, H.: Web-Based Map Generalization Tools Put to the Test: A jABC Workflow. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 171–181. Springer, Heidelberg (2014)
75. Noack, F.: CREADED: Coloured-Relief Application for Digital Elevation Data. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 182–195. Springer, Heidelberg (2014)
76. Respondeck, T.: A workflow for computing potential areas for wind turbines. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 196–211. Springer, Heidelberg (2014)
77. Scheele, L.: Location Analysis for Placing Artificial Reefs. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 212–224. Springer, Heidelberg (2014)
78. Kind, J.: Creation of Topographic Maps. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 225–234. Springer, Heidelberg (2014)
79. Holler, R.: GraffDok: A Graffiti Documentation Application. In: Lamprecht, A.-L., Margaria, T. (eds.) Process Design for Natural Scientists. CCIS, vol. 500, pp. 235–247. Springer, Heidelberg (2014)