An Open Grid Service Environment for Large-Scale Computational Finance Modeling Systems

Clemens Wiesinger, David Giczi, and Ronald Hochreiter

Department of Statistics and Decision Support Systems, University of Vienna

Abstract. In this paper we present the basic concepts of our complex problem modeling and solving environment based on a state of the art component architecture. We propose a system where components exist as instances of meta-components carrying relevant semantic information about the application problem realm. The implementation of the system follows the Open Grid Service Environment (OGSE) Service Stack, also discussed in this paper. A motivating workflow example from the field of computational finance is given.

1 Introduction

In the last decade the structure of applications changed from large monolithic pieces of code with some internal structuring to workflow applications, see [10]. Recently, Grid and Web Service based applications emerged, which provide the basis for the adaptation of our AURORA Financial Management System (see [13] for a discussion about prior implementations). In general, this system is a complex problem modeling and solving tool for large-scale financial decision models. Many efforts have been undertaken to bridge gaps between computer science and computational management science (for operational research approaches see e.g. [12][4], for distributed computing approaches see [6]). However, most of the available solutions for tackling problems in this area focus either entirely on low-level specialized problem formulations or on special optimization problem solutions. There is practically no abstract layer that provides a common framework in which components are interchangeable due to clear interface definitions and service descriptions.

In this paper we outline the nature of large-scale financial problems in general in section 2 and give an example for an typical problem in this area. Furthermore, we use this example to show how to apply a Grid environment to enhance the performance by exploiting intra-component and workflow parallelisms. The development of the Open Grid Service Environment architecture motivated by general considerations of component-based architectures for problem solving environments is discussed in section 3.

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3036, pp. 83–90, 2004.

2 Motivation for a Financial Problem Modeling and Solving Environment

In comparison to exact sciences like pure mathematics, decision science usually deals with incomplete information along with subjective models and even more subjective interpretation of solutions. Innovative models and solutions often exist but are spread throughout the scientific community. Therefore, the need for a common workflow platform for interchangeable components to reevaluate results and to extend large financial workflows in certain areas arises. Hence the proposed system should be capable of integrating and orchestrating different components for the realization of larger tasks, where maybe only a small part of the whole workflow is of interest to a specific researcher. In this manner components implemented by other people with different research focuses become useable and comparable.

It seems obvious, that such a flexible system needs higher level semantic descriptions of specific components. This is attained by defining meta-components and orchestrating these into a meta-workflow which gives a compound semantic description of what needs to be done and defines the steps to achieve this goal. Each concrete component is an implementation of a meta-component. The interchangeability of components derived from a meta-component arises from their common definition of their input and output structure.

Furthermore, it is important to emphasize that computational finance is an area where problems can be made arbitrarily complex in a computational sense. Like models from meteorology, chemistry, physics and material sciences models can be configured to consume all available computational power by making them finer and by that more realistic.

2.1 An Example from Computational Finance

As a prominent example for a computational heavy task, with possibly many subtasks, we present a specific multi-stage stochastic modeling and optimization problem, meaning that during the considered time period multiple consecutive decisions and possible recourse actions are modeled and subject to optimization. The goal of the optimization is to minimize the subjective risk of a financial portfolio, which is calculated from the return distribution, derived from historical development of the considered assets. The processing of this task can be broken into subtasks as described below. Each subtask exists as abstract description of the task to be performed and interchangeable actual implementations with the same input and output structure.

Below we present the meta-components that are involved in the workflow and also illustrate an example for an executable workflow formed from concrete implementations of the used meta-components. In contrast to the meta-workflow the concrete workflow is ready to be processed by workflow enactment mechanisms.

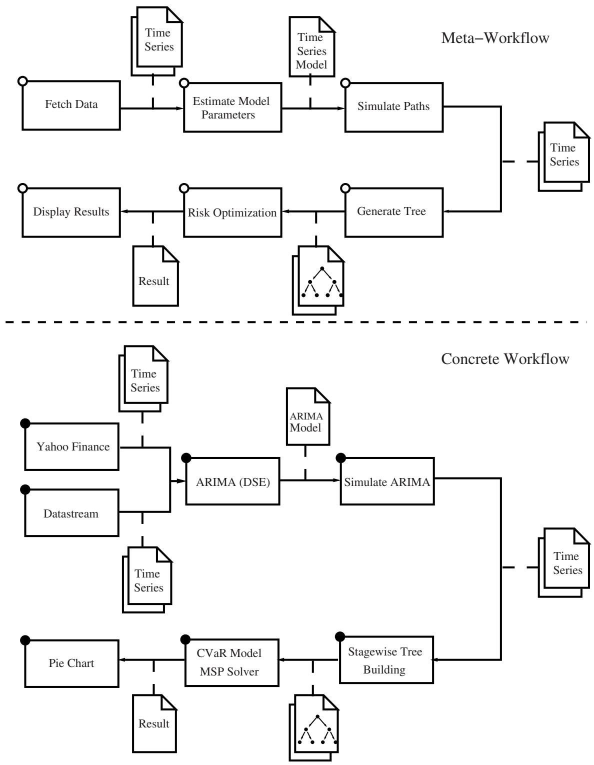

Fig. 1. OGSE workflow - multi-stage tree generation and portfolio risk minimization. Above the line the abstract meta-component workflow is described, while in the lower part the corresponding instanciation is shown. Every concrete implementation shown in the lower part has exactly one corresponding meta-description in the upper part. A rectangle with a circle in the upper left corner represents a meta-component, while a rectangle with a filled circle represents an implementation.

1. *Data fetching and converting components*: collect econometric (time series) data and convert to a suitable input format for consecutive steps. Differences between actual implementations arise from differences in data sources and data formats. All series are stored in the time series XML structure.
2. *Estimation components*: to capture inter-period dependencies in our data we use time series models to estimate the properties of our stochastic process. These components fit the data handed over from the previous step to a time series model and store the estimated parameters in a suitable format.
3. *Simulation components*: these components simulate a pre-defined number of trajectories according to the parameters of the model which are the output of the estimation step.
4. *Tree generation components*: in our example a tree is build out of the simulated trajectories. A tree can be viewed as a multi-dimensional filtration.

The output of a tree generation procedure is a XML tree object. We choose a method of stepwise merging of time series paths combined with a stage wise tree building procedure (see [5] and [7]).
5. *Optimization components*: solves e.g. a CVaR (Conditional Value at Risk) minimization problem (see [15]) with the tree structure from the tree generation step, and uses externally defined minimal expected return μ and confidence level α.
6. *Presentation components*: these components present results graphically in form of reports, tables, and charts.

The meta-component workflow and the described instanciation of the workflow are depicted in Fig. 1.

2.2 Parallel Performance Issues

It seems quite obvious that the finer the tree is, the better the true stochastic process is approximated. We could consider a reasonable realistic tree that models the time horizon as five stages where every node has five successors. The resulting tree consists of 3906 nodes with 3125 terminal nodes. If we double the amount of time steps and the number of successors in every node, we end up with 11111111111 nodes and 10000000000 terminal nodes, which accounts for an enormous increase in the computational complexity of the problem. However, it was shown in many publications that for this class of optimization problems parallel implementations can achieve a nearly linear speedup. Reported efficiency is usually larger than 90%, see [2] for a general overview of parallel optimization and [1] for multi-stage stochastic optimization, which is the common approach to solving large scale financial management problems. A first approach towards implementing multi-stage stochastic solvers on the Grid was successfully depicted in [11]. There is also scope for parallelization of other components besides the optimization itself, especially the tree generation methods, which are often computationally demanding, see e.g. [8] for a parallel clustering algorithm with super-linear growth, which can be used as the basis of many tree generation techniques.

Furthermore, it is possible to exploit not only intra-component parallelisms, but also concurrent work on different tasks of a workflow. The OGSE architecture is meant to provide the infrastructure for handling different types of parallelism in one common framework.

3 OGSE Architecture

3.1 OGSE Components

Figure 2 summarizes the ideas of [14] and [9] mapped to a component-based architecture which defines the ten main (software) building blocks of an open PSE.

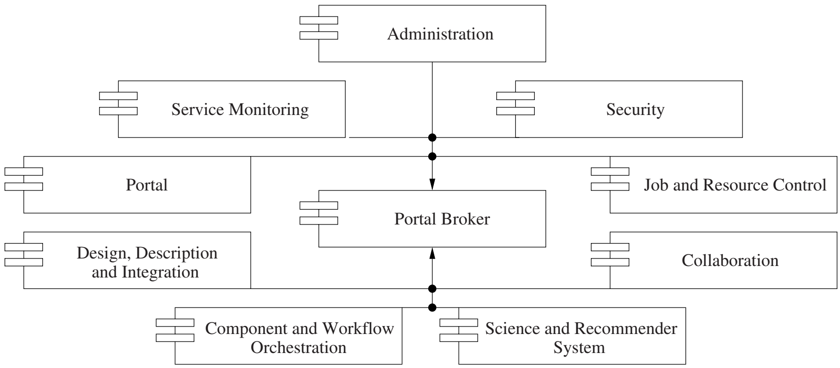

Fig. 2. OGSE Components for complex problem solving.

- *Portal:* portal contains user-centric presentations of the portal services. Groups of users have different roles such as scientific users, developers, and administrators with different views on the Grid-enabled problem solving environment.
- *Service monitoring:* the monitoring component provides facilities to compose monitoring operations, gather survey information about service activities and states, keep track of workflow execution, and include debugging and error recovery.
- *Administration:* these services enable members of privileged groups to perform maintenance and configuration of the problem solving infrastructure. Typical tasks are user management, service control, and service update.
- *Design, description and integration:* the system requires the usage and development of XML-based standards and specifications. Existing service standards, mainly lead-managed by the World Wide Web Consortium (W3C), Globus, and the Global Grid Forum (GGF), are exploited to provide a flexible (plug-in alike) service architecture for component integration (through semantic descriptions and rules).
- *Security:* the security component covers authentication, authorization, and confidentiality. Typical features are single sign-on, role-based user access, and data signing and encryption in the Grid environment.
- *Job and resource control:* every workflow consists of different jobs which have to be scheduled and are later submitted to several resources in the Grid system. According to the workflow orchestration, jobs have certain dependencies which have to be taken into consideration, when the workflow enactment is handled. Furthermore, resource monitoring and user role restrict the resource allocation and usage.
- *Collaboration:* is supported by user forums, Frequently Asked Questions (FAQs) and news bulletins, where relevant Grid and complex problem solving issues are discussed, common problems are listed and updates are announced. Collaboration services aim at creating a responsive scientific community,

where each user has an active role in the development and improvement of the complex problem solving environment.
– *Science and recommender system:* the recommender system advises researchers in the form of a knowledge base. Successful workflows are stored for knowledge mining. Along with workflow orchestration execution times and benchmarks are stored to compare different solutions and identify possible weaknesses in assembled workflows.
– *Component and workflow orchestration:* the orchestration workbench provides all accessible workflow components and predefined workflows that are stored in repositories along the Grid sites. A visual modeling desktop supports the user in discovering of workflow components and assembling of a specific workflow.
– *Portal broker:* the portal broker is the missing link in the complex problem solving architecture which integrates and connects the above named architectural components. The broker handles all messages between services and provides events to the user portal.

The above enumeration lists most of the issues considered important in the development of a PSE on the Grid. We understand that in a complete financial management system all of these mentioned points are nearly equally important and must be properly treated. As our research is mostly influenced by the financial application side we currently focus on the workflow orchestration, the building of the portal, the portal broker, and the definition of appropriate XML-based structures for service, problem, model, and data description. Other issues will be treated superficially in the next phase and extended in a later stage of the research project. If industrial partners start to use the system for consulting purposes other issues especially security and accounting gain importance.

3.2 OGSE Service Stack

The component collection introduced in section 3.1 provides a complex compound of highly distributed architectural services over an open PSE. To establish the distributed architecture, we propose an integrated service stack, the OGSE Service Stack (see Fig. 3), based on well defined and commonly used standards. The core building block of the OGSE Service Stack is the W3C Web Service Stack [3] which covers XML specification work in the workflow orchestration and enactment (workflow processing and monitoring services), service discovery, service description, and messaging. Semantics, also elementary covered by a W3C initiative, enhance the service core with PSE-specific descriptions in the area of problems, models, and (I/O) data. This vocabulary provides the foundation for an advanced arrangement of entities to an associative net, where constraints and relations are modelled with an assertional language into an ontology. The matching between a request for service through description data and the actual semantic data is provided by the matchmaking service. The OGSE API allows software developers to provide their own services.

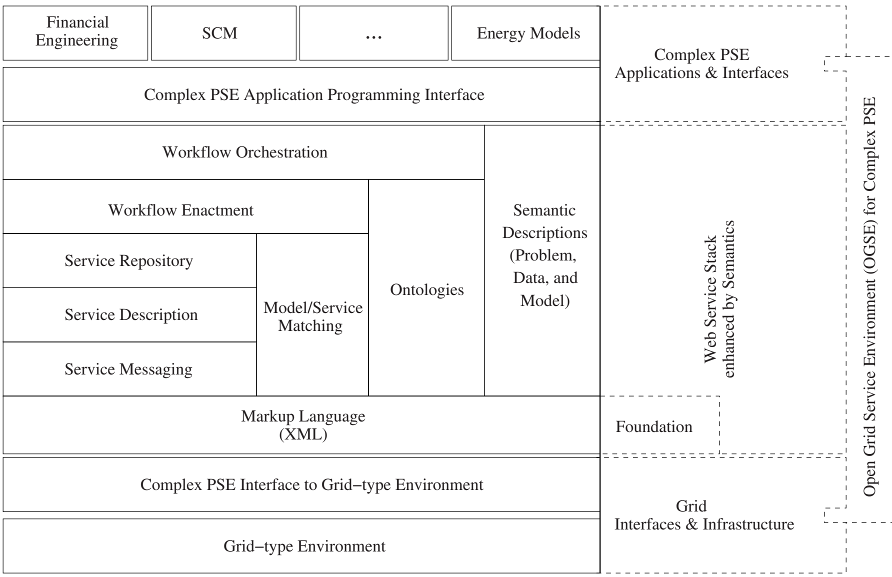

Fig. 3. OGSE Stack for complex problem solving applications.

The main goal of this stack is to establish an integration framework for a broad range of application areas with the main focus on an intelligent service discovery and workflow assembling in a computer processable way. The view on the service stack is application-centric from top down and Grid-centric from bottom up. The main ideas are the provision of

– *application-centric services*, that aim to cover the wrapping of existing legacy code, semantically describe facilities of the typical financial computation data, match the repository with the means of application data, and combine intelligent search features with the workflow orchestration.
– *grid-centric services*, that ensure compatibility requirements with computational grids (mainly covered by the Globus [16] initiative), workflow enactment on physical resources, and automatized discovery and allocation of computational resources.

4 Conclusion

In this paper we presented the basic concepts of our complex problem modeling and solving environment. This development grew out of the extension of the AURORA Financial Management System and has reached the level of being a unified framework for a broad range of application classes. The implementation of the core meta-workflow and the component architecture is based on the Open Grid Service Environment (OGSE). The meta-components within the system carry the semantic description which can be used for workflow composition and further (semi-)automatic component discovery and orchestration. A prominent example from the field of computational finance was briefly discussed to substantiate the relevance of further development on this system.

References

1. Blomvall, J. *A multistage stochastic programming algorithm suitable for parallel computing.* Parallel Computing. **29(4):** 431–445.
2. Censor, Y. and Zenios, S.A. *Parallel Optimization: Theory, Algorithms and Applications.* Oxford University Press. 1997.
3. Champion, M., Ferris, C., Orchard, D., Booth, D. Haas, H., McCabe, F., Newcomer E. *Web Services Architecture.* `http://www.w3.org/TR/ws-arch/`, W3C Working Draft. 8 August 2003.
4. Dempster, M.A.H, Scott, J.E. and Thompson, G.W.P. *Stochastic Modeling and Optimization using STOCHASTICS,* Draft 2002. Forthcoming in Applications of Stochastic Programming.,
5. Dupacova, J., Groewe-Kuska, N. and Roemisch, W. *Scenario reduction in stochastic programming: An approach using probability metrics.* Mathematical Programming **95A** (2003) 493–511
6. Ferris, M.C. and Munson, T.S. *Modeling Languages and Condor: Metacomputing for Optimization.* MetaNEOS Technical Report, 1998.
7. Gröwe-Kuska, N., Heitsch, H. and Römisch, W. *Scenario reduction and scenario tree construction for power management problems.* IEEE Bologna Power Tech Proceedings (A. Borghetti, C.A. Nucci, M. Paolone eds.), 2003
8. Iyer, L.S and Aronson, J. E. *A parallel branch-and-bound method for cluster analysis.* Annals of Operations Research **90** (1999) 65–86.
9. Laszewski, G., Foster, I., Gawor, J., Lane, P., Rehn, N. and Russell, M. *Designing Grid-based Problem Solving Environments and Portals.* Argonne National Laboratory, 2001.
10. Leymann, F. and Roller, D. *Workflow-based applications.* IBM Systems Journal **36**:1, 1997.
11. Linderoth, J. and Wright, S. *Decomposition Algorithms for Stochastic Programming on a Computational Grid.* Technical Report Computer Sciences Department, University of Wisconsin-Madison. 2002.
12. Kall, P. and Mayer, J. *SLP-IOR: An interactive model management system for stochastic linear programs.* Mathematical Programming **75** (1996) 221–240.
13. Pflug, G.Ch., Swietanowski, A., Dockner, E. and Moritsch, H. *The AURORA Financial Management System: Model and Parallel Implementation Design.* Annals of Operations Research **99** (2000) 189–206
14. Houstis, E. and Rice, J.R. *On the Future of Problem Solving Environments.* Department of Computer Sciences, Purdue University, 2000.
15. Rockafellar, R.T. and Uryasev, S.: *Optimization of Conditional Value-At-Risk.* The Journal of Risk **2(3)** (2000) 21-41
16. Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maguire, T., Sandholm, T., Vanderbilt, P., Snelling, D. *Open Grid Services Infrastructure (OGSI) Version 1.0.* Global Grid Forum Draft Recommendation, 6/27/2003.