Towards a Generic Visualization of Grid Resources

Timo Baur, Andreas Hanemann
Leibniz Computing Center
Boltzmannstr. 1
D-85748 Garching
Germany
{*timo.baur,hanemann*}*@lrz-muenchen.de*

Abstract

There are sustainable approaches on visualization in Grid environments, but there are no generic approaches for the visualization of a Grid environment itself. This work introduces a generic model and a set of services with the target of resource visualization and respect to the needs of Grid monitoring. The specified services are orchestrated as a visualization pipeline which is able to transform raw data about resources in consistent graphical views of the resources, generated according to their interconnections and relations. To achieve this, information and presentation models are integrated in the pipeline in a way that allows an easy adaptation of graphical representations to local requirements. The necessary functionality of services to conduct the transformations is also outlined. Finally, deployment scenarios are mentioned and a prototypical implementation is presented.

Keywords

Grid Monitoring, Resource Monitoring, Resource Visualization

1. Introduction

In previous years, Grid computing has attracted an increasing interest from academia and industry. While resource intensive tasks like climate simulations are able to provide a visualization of application problems, there is no established method for visualizing a Grid infrastructure itself and the resources constituting it. However, to ease the management of complex Grid infrastructures, appropriate methods for the visualization of Grid resources and their collaboration are highly desirable.

Latest Grid environments follow the paradigm of service oriented architectures (as adopted by the Open Grid Services Architecture OGSA [5]) and they make ample use of XML-descriptions to formalize resource status information. Such information models are often called "schema" or "resource descriptions". This paper introduces a service oriented approach for visualization, that is able to transform raw data about Grid resources into consistent graphical views, consistently imaging resource dynamics, interconnections and descriptions.

To achieve this, section 2 introduces a reference model, that divides the visualization

*Parts of this work have been funded by the German Federal Ministry of Education and Research under contract 01 AK 800 B.

process into several steps ("data states") and transformations. In sections 3, 4 and 5 the visualization process is analyzed and a definition of mathematical spaces for the states as well as the necessary functionality of transformation services is given and a framework for the integration of information models is described. Sections 6 and 7 finally mention a prototypical implementation at the Leibniz Supercomputing Center and related work. More information about further studies is given in the last section.

2. Reference Model

A basic process for data visualization was first described in [8] and formalized in the Filter-Map-Render Reference Model by Haber and Mc Nabb [6], describing a visualization process by conducting three necessary transformations on raw data: filter, map and render. In [3], the model was extended and the Data State Reference Model for information visualization was described, based on an evaluation of different existing visualization systems. Another terminology for the transformations (data transformation, visualization transformation, visual mapping transformation) was proposed and data was classified into four states (raw data, analytical abstraction, visualization abstraction, view), depending on its position in the process. Figure 1 shows this generic model for information visualization to the right. As this is a FIFO-like process, the data states and transformations are called a visualization pipeline.

For the application of the model to network or Grid management and monitoring, more ingredients are necessary: a formalism for the analytical and visualization abstraction states and transformation functions as described in [12] and an integration of management objects (respectively runtime representations of resource schema descriptions), their graphical representations and services conducting the transformations.

In the following sections, the data state reference model is used to deduce a service orchestration as shown in figure 1. Beginning at the bottom, raw data is provided by the data sources via the use of an API and/or application service and instantiated and mapped until a view can be generated that is ready to be delivered to the graphical user interface, shown at the top of the figure. MO (management object) and PO (presentation object) are data objects, described by a information model called a schema (in the case of MOs) or presentation schema (in the case of POs) and may themselves be realized as services. A relation service (under certain conditions already specified by the used SOA), a layout and a visualization service are necessary for the tasks specified below.

3. Data Transformation and Analytical Abstraction

As outlined in figure 1, raw data must first undergo a data transformation. Relevant data about the resources must be extracted and should be transformed into easy-to-handle data structures in a well-defined space. Such a space was proposed in [12], where an information space $IR = \{IM, IS\}$ in a visualization process is defined. It contains a set IM of information objects IO, and an information structure IS, keeping a set of relations between these objects. The information objects keep the information in their attributes A_n. For that, a function $attr\{IOn\} = \{A_1, A_2, ...\}$ is defined, delivering the attributes A_n of a set of objects.

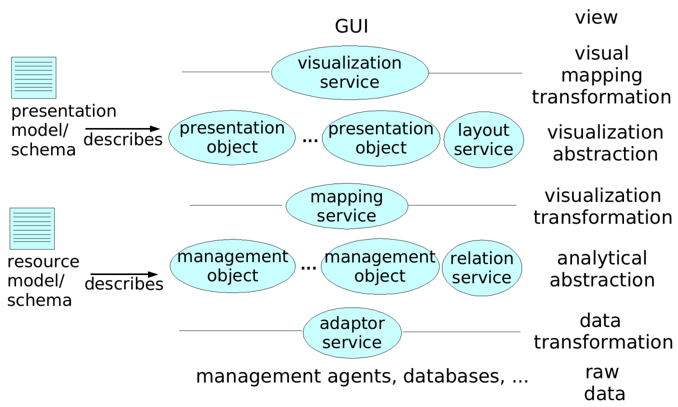

Figure 1: Information visualization pipeline, used terms and proposed services

In network and systems management, this approach to structure information of networks and systems is usually accounted for by incorporating different kinds of relations in information models. At instantiation, a part of the information model is realized in objects or data structures relating to the managed resource. In the Grid context, this could e.g. be the instantiation of a WS-Resource bearing all the data about the resource state, like e.g. processor load, or lifecycle information. To organize such raw data as management objects in an information space as described in [12], it is necessary to keep the relations separate, using a relation service and to provide an adaptor service or API to enable instantiation.

4. Visualization Transformation and Visualization Abstraction

For the next step, the transformation from analytical to visualization abstraction is described, where a description of graphical representation is applied and how tthe prestructured raw data contained in the analytical abstraction layer should be visualized. As shown in figure 1, this can be achieved by executing a visualization transformation, mapping the objects in the information space to their visual representations, using rules or heuristics. As done for the analytical abstraction state, it is possible to formalize a presentation space PR, consisting of a set PM of presentation objects and a set of functions: PR={PM,F}. According to [12] there exists a one-to-one mapping from IR to PR which is complete, unique and invertible. Thus, should the resources not only be monitored or visualized but also managed, the mapping process could also be conducted from presentation to information space. This functionality is dependent on the implementation of the services and objects and can be facilitated by the use of notifications. When a presentation object is changed, the corresponding (known by the mapping service) management object must receive a notification. A presentation schema must be defined for the presentation objects to specify the details of how and where every object from the information space should be visualized. Their implementation as a set of presentation objects defines the appearance and the visualization data of one or more objects from the analytical abstraction. For example, such a schema can define the two- or three-dimensional shape of a computing element and the coordinates of the shape in relation to its position in a window structure or scene graph. A mapping service should be able to instantiate a corresponding presentation object for every management object with an existing presentation schema and keep the set of presentation objects consistent with the lifecycle of the corresponding management objects in the state of analytical abstraction.

After mapping, a set of unlocated presentation objects is instantiated and in place. Now, location information has to be set, to undertake an architectural organization of the presentation base and enable the next transformation to visualize the objects. This is the purpose of the layout service which is essential to structure the objects in the surrounding space by using the relations stored in the relation service. Such a service is only one imaginable function on the PR, but it is essential for subsequent scene composition and rendering. More than one layout service with different independent layout policies can be defined.

5. Visual Mapping Transformation

When data in the state of visualization abstraction is given, it can finally be transformed into a complete graphical view by applying a visual mapping. For 2D-visualizations, a structuring of the scene to the model of the used graphics engine or API (e.g. Java Swing) must be applied; for 3D-visualizations, this is usually done by generating a scenegraph (e.g. for Java3D or VRML), defining a camera position and rendering the scene. Appliance of this visual mapping is the task of the visualization service, that will provide an appropriate presentation model for the graphics engine.

6. Prototype

To show that the concept is usefull, a prototype called NEVE Network Visualization Environment [1] was constructed and evaluated as an integrated plattform in a component container based on JMX (Java Management Extensions). Up to 1600 entities in an IP network have been visualized, while providing a platform for a three-dimensional visualization and management of the objects (with the possibility to connect adaptors for the management of underlying devices) at the same time. The GUI also provides navigational functionality through the visualization. For a small number of resources, the prototype is able to view changes in the information base very quickly. Short durations of state (e.g. the existence of an HTTP-Traffic flow) could be visualized down to an intervall of 25-30 ms on a medium performance desktop computer.

For the tests, all services have been on the same machine. Because of the service paradigma, performance should be improved further depending on the involved hardware platforms and the distribution of GUI and services.

7. Related Work

There are already a few approaches for visualization in Grid computing. The Grid visualization kernel [7] is a powerful application for data visualization, even allowing parallel visualization pipelines, and it is integrated as a part of a distributed service-oriented middleware infrastructure. For the middleware Naregi [9], generalized Grid visualization services have been proposed and implemented as a plugin for the middleware Unicore. Both approaches do not consider an integrated global view of data published by heterogenous and distributed resources in the Grid environment, but could fit quite well for conducting the visual mapping transformation in the upper part of the introduced model. The project gViz [11] points in an interesting direction by an approach “to study the potential of XML for visualization in the context of Grid computing” [2]. While the work concentrates on the definition of XML presentation models for the visual mapping transformation, the visualization transformation, situated on a lower level of the process, i.e. the mapping from resource descriptions to presentation schemata is not accounted for. In contrast to these projects, this paper outlines and analyzes the full path, data about Grid resources must take until a view can be seen by the user and introduces localized mechanisms for the mapping of the schema-based information of the management object to presentation information about how a single object should be visualized. This is essential for a granular visualization of heterogenous and distributed Grid resources and thus a Grid environment itself. In addition, the data objects used in this approach may be active and - being themselves manageable - they may possess information about themselves. Their functionality may be extended to platform- (i.e. middleware-) wide functionalities like notification or security facilities. In this way, a dynamic and flexible view to the properties and functions of the real resource the data is relating to and even their management can be provided.

8. Conclusion and Future Work

Further research is done in applying the concept to Grid resource monitoring and management. For this purpose, an analysis of heterogenous sources of resource data is conducted and methods to structure and discover relational Grid resource topology information are investigated. These can e.g. be based on WS-ServiceGroups as used in the monitoring framework of the Globus Toolkit, MDS4 [10], the monitoring framework R-GMA [4] or others.

The specified services can be deployed in a variety of ways, e.g. as a local and centralized integrated platform, a centralized service implementor implementing mapping service and upper layers, a single centralized visualizer or as a completely distributed setup in platforms as described by OGSA. Amongst other factors, the ideal deployment is depending on the underlying service, middleware and network infrastructure and the distribution and kind of resources to be monitored and visualized. Appropriate measurements and guidelines for scalable and performant service deployments should be found with respect to issues like the required actuality of data in the several states up to and including the view of the end user as well as polling and notification intervalls. Further research must be done to find appropriate measurements and guidelines for scalable and performant service deployments with respect to these issues. These are general problems of Grid monitoring infrastructures and most of them are not yet sufficiently researched.

Acknowledgement

The authors wish to thank in particular Prof. Dr. Gabrijela Dreo Rodosek and Dr. Victor Apostolescu and the members of the Munich Network Management (MNM) Team, directed by Prof. Dr. Heinz-Gerd Hegering, for helpful discussions and valuable comments on previous versions of this work. Its web-server is located at http://www.mnm-team.org

References

[1] T. Baur. Entwurf einer architektur zur integration von netzplanungs- und managementwerkzeugen in eine vr-umgebung. Master's thesis, Technical University Munich, 2002.

[2] Ken W. Brodlie, Jason Wood, David A. Duce, Julian R. Gallop, D. Gavaghan, Michael Giles, S. Hague, Jeremy Walton, M. Rudgyard, Brian Collins, J. Ibbotson, and Alan Knox. Xml for visualization. In *EuroWeb*, 2002.

[3] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *INFOVIS*, pages 69–76, 2000.

[4] A. W. Cooke, A.J.G. Gray, W. Nutt, J. Magowan, M. Oevers, P. Taylor, R. Cordenonsi, R. Byrom, L. Cornwall, A. Djaoui, L. Field, S. Fisher, S. Hicks, L. Leake, R. Middleton, A.J. Wilson, X. Zhu, N. Podhorszki, B.A. Coghlan, S. Kenny, D. O'Callaghan, and J. Ryan. The relational grid monitoring architecture: Mediating information about the grid. *J. Grid Comput.*, 2(4):323–339, 2004.

[5] I. Foster, H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell, and J. Von Reich. Open grid services architecture, version 1.0. 2005.

[6] R.B. Haber and D. A. McNabb. Visualization idioms: A conceptual model for scientific visualization systems. In *Visualization in Scientific Computing*. 1990.

[7] D. Kranzlmüller, P. Keinzlreiter, and J. Volkert. Grid-enabled visualization with gvk. Technical report, European Across Grids Conference, 2003.

[8] Bruce H. McCormick, Thomas A. DeFanti, and Maxine D. Brown (ed). *Visualization in Scientific Computing*. ACM SIGGRAPH, New York, 1987.

[9] K. Miura. Grid computing and scientific visualization. Presentation, 2004.

[10] J. Schopf. Mds4: The gt4 monitoring and discovery system. Technical report, Globus Alliance, 2005.

[11] Jason Wood, Ken Brodlie, and Jeremy Walton. gviz - visualization middleware for e-science. In *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 82, Washington, DC, USA, 2003. IEEE Computer Society.

[12] V. Wünsche. Eine begriffswelt für die informationsvisualisierung. *Rostocker Informatik Berichte*, 21, 1998.

Biography

Timo Baur graduated in computer science at the Technical University Munich in 2002. He is a member of the MNM Team and PhD candidate at the Ludwig-Maximilian-University in Munich, Germany. He is working in a Grid monitoring project for the german D-Grid initiative and his research interests include Grid computing, communication networks, resource and service management as well as cyberanthropology and information visualization.

Andreas Hanemann received a diploma degree in computer science from the University of Karlsruhe (TH), Germany. He is a member of the MNM team and PhD candidate at the Ludwig-Maximilian-University in Munich, Germany. He is working in a network monitoring project at the Leibniz Supercomputing Center that deals with establishing a monitoring infrastructure for several research backbone networks. His research interests include service management, service fault management, and service-oriented event correlation.