A Grid Service-Based Active Thermochemical Table Framework

Gregor von Laszewski[1], Branko Ruscic[2], Patrick Wagstrom[1], Sriram Krishnan[1], Kaizar Amin[1], Sandeep Nijsure[1], Sandra Bittner[1], Reinhardt Pinzon[2], John C. Hewson[3], Melita L. Morton[2], Mike Minkoff[1], and Al Wagner[2]

[1] Mathematics and Computer Science Division
[2] Chemistry Division
Argonne National Laboratory 9700 S. Cass Avenue, Argonne, IL 60439, U.S.A.
[3] Sandia National Laboratories, Livermore, CA 94551-0969, U.S.A.
gregor@mcs.anl.gov
Third International Workshop on Grid Computing, Baltimore, MD, 18 Nov. 2002

Abstract. In this paper we report our work on the integration of existing scientific applications using Grid Services. We describe a general architecture that provides access to these applications via Web services-based application factories. Furthermore, we demonstrate how such services can interact with each other. These interactions enable a level of integration that assists the scientific application architect in leveraging applications running in heterogeneous runtime environments. Our architecture is implemented by using existing infrastructures and middleware, such as Web services, the Globus Toolkit, and the Java CoG Kit. We test our architecture on a thermochemistry application that provides a number of requirements, such as batch processing, interactive and collaborative steering, use of multiple platforms, visualization through large displays, and access via a portal framework. Besides the innovative use of the Grid and Web services, we have also provided a novel algorithmic contribution to scientific disciplines that use thermochemical tables. Specifically, we modified the original approach to constructing thermochemical tables to include an iterative process of refinement leading to increased accuracy; we are now implementing this approach. We have designed a portal for accessing the set of services provided, which include the display of network dependencies between the reactions a chemist may be interested in and interactive querying of associated species data.

1 Introduction

The study of energy changes that accompany chemical reactions and changes in the physical states of matter is referred to as *thermochemistry.* The knowledge of thermochemical stability of substances is central to chemistry and critical in many industries, since chemical reactions are ultimately governed by thermochemistry. Hence, thermochemistry finds applications in other disciplines such as earth science and engineering, helping scientists to better understand processes such as climate and combustion [18, 11] and thus predict and verify them to a high degree of accuracy.

Until now, the thermochemical data necessary for such calculations has been available only in static table form, and the algorithms to derive accurate model descriptions have been too imprecise to deal with the complex chemical reactions encountered in

state-of-the-art laboratory experiments or observed in nature. Our goal is to improve this situation by delivering innovative algorithms to the scientists through an advanced collaborative environment.

Novel modalities of deriving new scientific results can be stimulated by enabling a collaborative environment in which scientists can publish and share their results with others, perform sophisticated calculations that are otherwise not feasible, and integrate newly developed algorithms.

The Grid [27, 9, 10] can provide the basic middleware infrastructure for bootstrapping this type of sophisticated collaborative environment. The Grid allows scientists to collaborate even though their resources may be controlled by different domains; access to these resources is enabled through the use and creation of "virtual organizations".

In this paper, we show how we provide advanced services that can be accessed collaboratively. Their integration as part of a workflow process enables the creation of services that can be easily reused by the community. Scientists are then in a position to concentrate on the science, while application developers can focus on the delivery of services that can be assembled as building blocks to create more elaborate services.

Our paper is structured as follows. First we give a short introduction to the problem domain and the terminology used in thermochemistry that is directly related to the work we perform. We analyze a current process to derive thermochemical tables, one of the most elementary building blocks in thermochemistry. Next we provide an improved technique for increasing the accuracy of this process. We introduce a scenario where our algorithm and the repeated use by the community will result in a highly accurate and elaborate thermochemistry table database. We outline our service-oriented architecture and discuss how services such as security, data transfer, registration, and scheduling assist in assembling such a sophisticated collaborative environment. We conclude our paper by summarizing the current state of the project and listing opportunities for further research.

2 Basic Thermochemistry

In this section we provide a minimal introduction to basic thermochemistry that is necessary to understand the services and scenarios presented later in this paper. A more complete account of the thermochemical development behind the concept of the active thermochemical tables will be published in the near future.

Elementary to the discipline of thermochemistry is *enthalpy* (ΔH_f^o), which refers to the value of energy of a system when it is at constant pressure. The enthalpy relationships involved in examining thermochemical equations are easily visualized by means of enthalpy diagrams, such as that shown in Figure 1.

In this diagram the equations are expressed as a graph with horizontal lines representing different values of the enthalpy. Typically the differences between these values are determined experimentally or can be derived by using the thermodynamic laws from other enthalpy values. Values obtained from experiments, however, may contain errors (not shown in the diagram). Changes in the enthalpy are visualized by the distance between the lines. Based on the changes performed, different intermediate states (chemical species) may occur during the transition from one to the other final state. Thus, it is

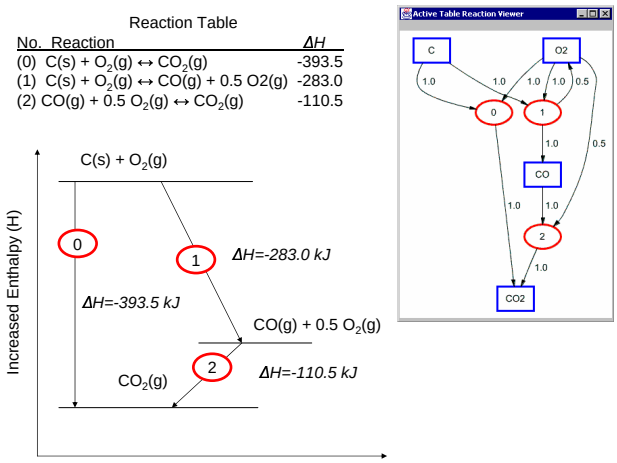

Fig. 1. Enthalpy diagrams and thermochemical reaction tables.

natural to visualize the transition with directed edges between the states. An alternative graph is displayed on the upper right-hand corner that emphasizes the possible states in which a chemical species (i.e., an ensemble of chemically identical molecular entities that can explore the same set of molecular energy levels on the time scale) can occur.

In traditional thermochemical tables the enthalpies of formation (ΔH_f^o) are developed with the help of an elaborate *sequential process*. In each step a new species is added while all ΔH_f^o values determined in previous steps are frozen. The enthalpy of the new species is determined at one temperature T from interconnecting measurements that are limited to those species already defined in previous steps. The temperature-dependent functions, $C_p^o, S^o, (H_T^o - H_o^o), \Delta H_{fT}^o, \Delta G_T^o$, are developed by determining the partition function $Q, lnQ, T\partial(lnQ)/\partial T$, and $T^2\partial^2(lnQ)/\partial T^2$ from the species-specified quantities and the newly selected single temperature enthalpy. (For more information about these terms, please see [20].)

The sequential process follows a *standard* order of chemical elements: $O \rightarrow H \rightarrow halogens \rightarrow noblegases \rightarrow chalcogens \rightarrow pnicogens \rightarrow carbonperiod \rightarrow etc.$ For every chemical element introduced, the sequence starts at the standard state for that element, for which $\Delta H_f^o = 0$ by convention. However, enthalpies of formation have complex hidden dependencies. These dependencies are backwards traceable, albeit with considerable manual effort, and in practice are not forwards traceable at all.

This sequential approach has several disadvantages. In particular, it results in ΔH_f^o and error bars that do not properly reflect the global relationships implied by the species-interconnecting data used. The values and error bars reflect, at best, only local relationships to nearest neighbors. A cumulative error is introduced based on the frozen enthalpies in previous steps. Furthermore, the hidden relationships in conventional tables produce thermodynamic tables that are static in its nature. Proper update with new knowledge is nearly impossible because forward relationships are nontranspar-

ent; hence, updating one species may improve things locally but increase the global inconsistencies.

We present a new approach, based on active tables, that circumvents these disadvantages.

3 Active Thermochemical Tables

The active table approach treats the information in the species-interdependent data from the viewpoint of a thermochemical network [20, 19]. Every vertex (node) of this network represents the enthalpy of formation of one species. The species-interdependent data define the topology of the network graph by providing the edges (links) in the network. Competing measurements provide multiple (parallel) links in the network. The relational information defining the topology maps onto a set of equations, with the enthalpies of the involved species as unknowns, the stoichiometry of the reactions defining the coefficients, and the measurements and their error bars providing the free elements. Since the number of equations regularly exceeds the number of unknowns, the system is overdetermined. The best solution to the network is obtained in two steps.

The first step statistically analyzes the network with the goal of checking the error bars of individual data items for consistency. All inconsistencies are identified and proposals for their resolutions generated. The resolutions generally proceed through incrementing the error bars of the offending data items to their statistically indicated values, and the analysis is repeated. During subsequent steps the error bars of the offending data items usually oscillate up and down until self-consistency across the whole network is reached. The analysis is carried out in both unweighted and error-weighted space.

The second step occurs once the values and error bars of data items achieve consistency across the whole network. This step consists of finding simultaneously the optimal solution to all nodes by χ^2 minimization (in error-weighted space). The species-specific data is used to prepare the network for analysis by re-expressing all data items as reaction enthalpies at one common temperature. Once the optimal solution to the network has been found, the species-specific data is used to calculate the partition functions and hence develop the temperature dependence of the standard thermochemical functions.

We note that, as opposed to the simultaneous solution described above, in the traditional sequential approach the nodes are solved one at a time in a prescribed sequence. Each of the steps corresponds to selecting a particular path in the network leading from solved nodes to the next node, and ignoring all other possible paths. Both the fact that an overdetermined system of coupled equations is solved for one unknown at the time and that some of the equations are ignored contribute to the lack of global consistency in traditional tables.

4 Benefits of the Active Table Approach

Unlike the quantities found in a traditional table, the thermochemical quantities (and their error bars) obtained from the Active Table properly reflect the globalism of relationships implied by the underlying thermochemical network. All values and error bars

are consistent in a global sense. An active table allows rapid update with new measurements (and possibly calculations) by globally propagating the new information through the table. Quality and integrity of the table are protected throughout the updates by error analysis, which runs in both directions: The error bar of the new experiment may shrink error bars in the table; however, the error bars of other experiments in the table might challenge, by means of the statistical analysis discussed above, the error bar assigned to the new experiment. In addition, an active table allows "what-if" tests. Such tests provide a critical evaluation of the tested data and its impact on prior thermochemical knowledge or, if the tests correspond to a new experiment, provide feedback on sensitivity of the network to various measurements. The approach has also potential to become an interesting learning and education tool. An active table can provide a ranked list of links that are missing or weak from a statistical viewpoint; in particular, it can provide pointers to the most useful new experiments or calculations.

5 Application Requirements

We have performed an initial requirements analysis that identified a number of important basic use cases that must be provided by our architecture. These use cases include

- the calculation of an active thermochemical reaction,
- the visualization of a set of active thermochemical reactions as a graph,
- the polynomial fitting (used in subsequent modeling) of a function based on data obtained either with standard or with active thermochemical tables from a variety of input sources,
- the display of the polynomial fitted functions, and
- the query of data needed for the polynomial fitting and the active tables.

In the next section we extend our requirements list beyond individual scientific applications needs and concentrate more on the modalities of using these applications in a shared and collaborative environment.

6 Grid Requirements

To decide on a computational environment that supports our proposed algorithmic solution, we first analyzed the requirements based on the modality of the scientific research to be performed. We identified the following requirements:

Collaborative environment that supports the interaction among scientists in geographically dispersed locations.

Secure environment that protects from the loss of intellectual property and allows restricted access to the data and compute resources.

Standardized environment that enables the scientist to use the tools in a straightforward fashion.

Adaptive environment that is flexible to future changes based on hardware and software.

Dynamic environment that allows the creation of transient services to enable adhoc collaboration and use of other application services.

These requirements are shared with many other scientific disciplines, and a large amount of research has been performed in the past few decades to develop frameworks that support such requirements.

We decided to base our architecture on a framework that is centered on the concept of the Grid. The Grid enables flexible resource sharing among a collection of resources that is maintained as part of different administrative domains. Middleware such as the Globus Toolkit can provide the foundation for an implementation of our architecture. Additionally, we need to develop advanced application specific services that build on basic Grid services so that scientists can use standardized Grids as defined by the Global Grid Forum [4].

Besides the integration of Grid standards we also need to take into account the availability of commodity tools and frameworks such as Web services that enable a bridge to commercially available middleware, thus simplifying our implementation. Examples of successful bridging initiatives include the Globus Project through the Java Commodity Grid Kit and, more recently, the Open Grid Service Architecture, which is currently under development. Drawing on this rich experience, we prototyped an architecture that can be supported by these commodity-integrating technologies.

Further analysis of our problem domain revealed that it is beneficial to build the framework based on a service-oriented architecture. Such an environment includes flexible design while still being able to integrate sophisticated security solutions. Additionally, we can design services that interact with each other and may operate at geographically dispersed locations. We have identified within our project the need to deliver the following services:

Grid Broker Service to deal with large numbers of calculations that are involved with future large-scale reactions and their real-time requirement for allowing interactive use [23, 2].

Grid Workflow Service to enable the interplay of Grid services through workflow descriptions [24, 25].

Grid Execution Factories to enable the execution of programs in a Grid while instantiating them in a hosting environment and making their results accessible to other services (the Globus Toolkit provides such services in Java and C) [24, 14].

Grid Monitoring Service to monitor the state of the hosting environment so that feedback to Grid services is provided, enabling the environment to react to state changes [26].

Grid Migration Service to be able to migrate services and jobs executed with a Grid Execution Factory Service to a location that is better suited, based on performance and quality of service descriptions and policies [28].

Grid Logging Service to log and checkpoint services in order to enable migration and fault-tolerant behavior.

Grid Self-Healing Service to determine how and when it is necessary to change the dynamically instantiated Grid workflow applications (including preventive measurements such as service replication, service migration, service checkpointing, and service monitoring).

Collaborative Steering Service to collaboratively create data, thoughts, and ideas that will lead to new scientific findings.

Furthermore, we require a simple portal to interact with this sophisticated environment, so that scientists may concentrate on the science and not the environment [27]. Although many more services are needed, we decided to restrict our initial prototype on these sets of services.

7 Architecture

We have identified that a service-oriented architecture with a discovery and binding mechanism can be used to deal with the dynamically changing nature of our collaborative environment. This architecture must enable to connect several functional services that perform the tasks demanded by the application. We depict these application specific services in Figure 2.

The scientist can interact with our computational programs via portal mechanisms. The functions currently supported by our portal are polynomial fitting of data based on thermochemical data provided in a standard format such as JANAF [17, 22, 29], the query of thermochemical data for species, the creation of a database based on species data, the calculation of an active table based on a number of reactions (currently under development), and the graphical display of the polynomial fitted data and the reactions.

The elegance of our architecture is based on the use of the service model that allows us to be flexible in many ways. First, we provide many of the algorithmic functions as services that can be placed in a geographically dispersed environment. Thus it is possible to maintain changes to the original algorithm by the application specialist, while at the same time minimizing the effort for reuse in a collaborative environment. Second, we are able to integrate new services into this architecture to extend it while being open to future requirements. Thus, we have created an architecture that is open and allows for expansion during the course of its development. Third, we are able to replace parts of our architecture with newly developed services, providing a customized functionality to disparate user communities.

In our first prototype we provide a Swing portal, as well as a Web-based portal based on the Jetspeed framework [5].

We have chosen intuitive names for our application-oriented services so that their functionality may be effectively communicated. Some of our services are as follows:

Polynomial Fitting Service that performs the polynomial fitting of data based on standard thermochemical tables such as JANAF

Active Table Service that performs the active table algorithms and the χ^2 minimization of the systems of linear equations obtained from a reaction table. This service is composed of two services:

- **Query Service** that returns information about the chemical species
- **Reaction Service** that allows the graphical display of the reactions to be analyzed

Plotting Service that allows the creation of two-dimensional data plots

Graph Service that allows the creation of two-dimensional visualization of augmented graphs (currently using the GraphViz engine [12, 13])

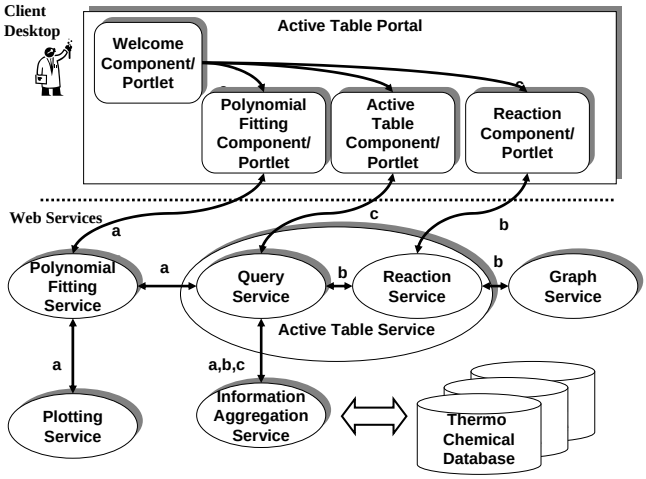

Fig. 2. The application-specific services to enable the scientist to access a convenient portal interfacing exposing active thermochemical tables, polynomial fitting of JANAF data, and their visualization. The letters a, b, and c refer to the corresponding use cases as explained in the text.

Information Aggregation Service that allows querying multiple databases maintained in a distributed fashion. A caching mechanism minimizes the search latency for frequently asked queries.

8 Example Use Cases

Based on our instantiation of our architecture on a hosting environment, we provide a series of screenshots to illustrate the state of our implementation.

Use Case A: Requesting a polynomial approximation of the thermochemical characteristics of argon. The user starts up the polyfit client on his desktop and requests a mathematical approximation for the thermochemistry properties of a given species, for example, argon. The polyfit client then opens a communication channel to a polyfit Web service to send the request. To process the request, the polyfit service needs the basic thermo data for argon, which is requested and returned from the Active Table Service. Having the required information, the polyfit Web service calls the non Grid-aware FORTRAN application to process the table information and produce coefficients for equations. These equations are passed to the plotting Web service to generate a graphic image. This graphic image is then returned to the polyfit client, and the data can be viewed on the users terminal. The workflow for the components and services that are involved in this use case are indicated with the letter *a* in Figure 2. The portal interface is shown in Figure 3.

Use Case B: Visualizing a reaction graph and obtaining information on the species in the graph. The user starts up the reaction visualization client and submits a reaction

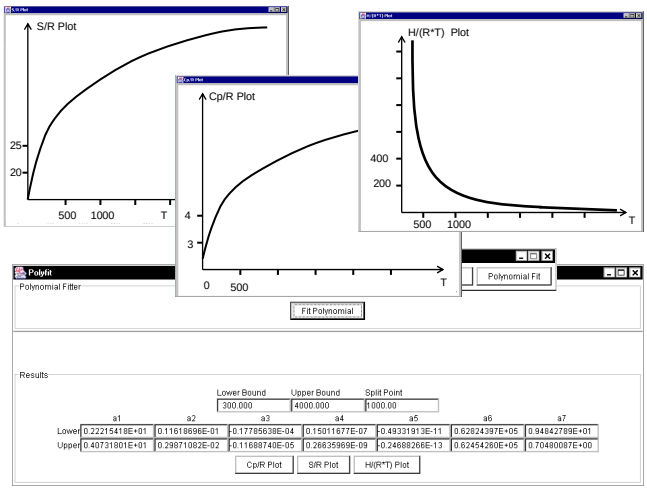

Fig. 3. The polynomial fitting of data based on JANAF tables.

file to this client. A reaction file can be as simple as one chemical reaction, or it can be hundreds of reactions all working together. The client then passes the information on to the graph service. The graph service takes this reaction file and parses it. It then calls some non-Grid-aware graphing service such as Dot [12] to create the graph. In order to provide more information on each of the species in the graph, the graph service also connects to the active table service for each of the species in the reaction file. This information is then passed back to the client and then the user for examination. The workflow for the components and services that are involved in this use case are indicated with the letter *b* in Figure 2. The portal interface is shown in Figure 4 with the reactions as depicted in Table 1.

Table 1. A thermochemical reaction table

Equation	Enthalpy (kJ/mol)	Uncertainty
$O_2(g,0) \Leftrightarrow 2\,O(g,0)$	493.579	0.179
$H_2(g,0) \Leftrightarrow 2\,H(g,0)$	432.071	0.012
$\frac{1}{2}O_2(g,298) + H_2(g,298) \Leftrightarrow H_2O(l,298)$	-285.830	0.040
$\frac{1}{2}O_2(g,298) + H_2(g,298) \Leftrightarrow H_2O(l,298)$	-285.795	0.040
$\frac{1}{2}O_2(g,298) + H_2(g,298) \Leftrightarrow H_2O(l,298)$	-285.850	0.330
$H_2O(l,298) \Leftrightarrow H_2O(g,298)$	44.004	0.002
$H_2O(l,298) \Leftrightarrow H_2O(g,298)$	44.016	0.010
$H_2O(l,0) \Leftrightarrow H_2O(g,298)$	-2.093	0.001
$OH(g,0) \Leftrightarrow O(g,0) + H(g,0)$	423.717	0.179
$OH(g,0) \Leftrightarrow O(g,0) + H(g,0)$	424.076	1.196
$H_2O(g,0) \Leftrightarrow OH+(g,0) + H(g,0)$	1748.101	0.772
$H_2O(g,0) \Leftrightarrow OH+(g,0) + H(g,0)$	1748.207	0.338
$H_2O(g,0) \Leftrightarrow OH+(g,0) + H(g,0)$	1748.256	0.193
$H_2O(g,0) \Leftrightarrow OH+(g,0) + H(g,0)$	1748.101	0.482
$OH(g,0) \Leftrightarrow OH+(g,0)$	1255.947	0.024
$OH(g,0) \Leftrightarrow OH+(g,0)$	1255.274	0.965
$OH(g,0) \Leftrightarrow OH+(g,0)$	1254.309	9.649
$H_2O(g,0) \Leftrightarrow OH(g,0) + H(g,0)$	492.275	0.060
$\frac{1}{2}O_2(g,0) + \frac{1}{2}H_2(g,0) \Leftrightarrow OH(g,0)$	37.082	0.670
$H_2O_2(l,298) \Leftrightarrow H_2O_2(g,298)$	47.950	4.400
$H_2O_2(l,298) \Leftrightarrow H_2O_2(g,298)$	51.920	0.150
$H_2O_2(l,298) \Leftrightarrow H_2O_2(g,298)$	47.510	3.100
$H_2O_2(l,298) \Leftrightarrow H_2O_2(g,298)$	51.750	0.160
$H_2O_2(l,298) \Leftrightarrow H_2O_2(g,298)$	52.200	10.000
$H_2O_2(l,298) \Leftrightarrow H_2O_2(g,298)$	51.925	0.073
$H_2O_2(g,0) \Leftrightarrow H_2O_2(g,298)$	-5.990	0.001
$H_2O_2(g,0) \Leftrightarrow 2\,OH(g,0)$	203.985	0.041

Use Case C: Requesting the thermo table for carbon (graphite). The user starts an active tables client and enters a search string into the client, in this case *Graphite*. This data is then submitted to the active tables Web service. If the service does not contain

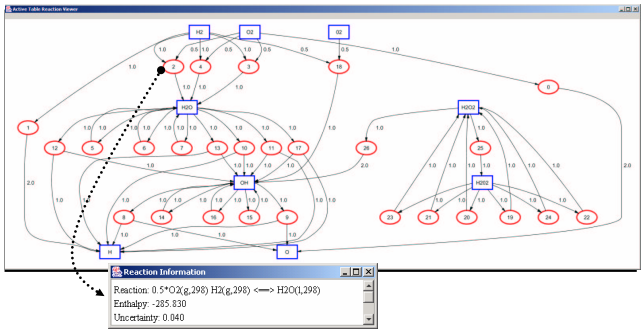

Fig. 4. A reaction graph visualization component.

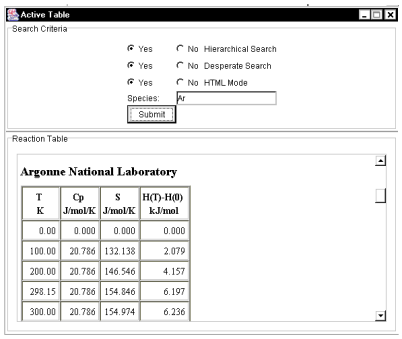

Fig. 5. Querying the species dictionary.

enough information to process the request, it may request more information from a WebDAV [15] server or other Web services. That new information is returned to the active table services, where it is combined with the data obtained from the active tables program. This is then passed back to the client and then the user. The workflow for the components and services that are involved in this use case are indicated with the letter *c* in Figure 2. The portal interface is shown in Figure 5.

Use Case D (currently in planning): Using the active table in educational outreach projects. Because we use standard Grid security infrastructure [8] and the ability to flexibly assemble our services, it will be possible to create a customized portal access through the *Access Grid* [7]. This will allow us to share and display our interface through Active Murals among a set of participating institutions in an educational setting. The ongoing DOE-sponsored SciDAC projects [21] will enable the easy integration of our services in the near future, when we expect that Web services technologies

are adopted within the Grid community. A cartoon of this use case is depicted in Figure 6.

9 Implementation

Our implementation follows the strict separation of backend services from rendering services as depicted in Figure 2. On the backend, we provide an application factory service, which is capable of launching non-Grid-aware applications, after initializing them with parameters and other command line arguments. We use such a service to export complex FORTRAN code (written by the chemists) as a Web service, without having to modifying it. In the future, clients will be able to retrieve a handle to such a service using standard mechanisms (e.g., UDDI [6]) and will be able to access the scientific codes, as shown in Figure 7. We prototyped a generic tool that exposes any non-Grid-aware code as a Grid service. This tool accepts the interfaces that the service should export (using some standard format, e.g., IDL, WSDL, Java interfaces) and the bindings from each method in such an interface to the actual calls to the non-Grid-aware application, and generates the required glue code to expose it as a service. The implementation is along the lines of the WSDL-to-Java converter tool provided by Axis [1] and, in fact, involved a slight modification of the code generation provided by Axis to suit our needs.

The graph service is also exported as a Web service, for the same reasons. The reactions that are fed into the graph service, in a format convenient to chemists, are then parsed and converted to a canonical XML format by using Castor [3], which is a data-binding framework for Java. The standard XML representation with corresponding Java bindings was a logical choice, as we can then plug in many different types of visualizers at the backend, without having to worry about conversions from the chemical format to the formats expected by the graphical packages. Currently, our visualization engine uses Dot to convert the XML representation into an SVG format, but we anticipate having several ways to visualize these graphs. The XML format also lends itself well to being transferred, and can be visualized locally, if need be, to save bandwidth.

Through the use of SWING applications and a Jetspeed portal we have shown that our services can be reused by a variety of frontend systems.

10 Future Developments

We will enhance our framework on multiple levels. First, we will enhance our Grid environment with more advanced services such as the ones listed in Section 6. These services allow us to create a sophisticated Grid environment that enables the creation of adaptive and self-healing Grid services. As our services can be assembled with each other using the Grid Workflow Service [16], we can provide application- and non-application-specific services with features currently not provided by the current generation of Grid software. Figure 8 shows an example of how we will use these services for reliable and flexible brokering of jobs in Grids with quality-of-service assurances. The goal is to investigate the design of a highly available service that can adapt itself to the disruptive nature of the Grid and the Internet.

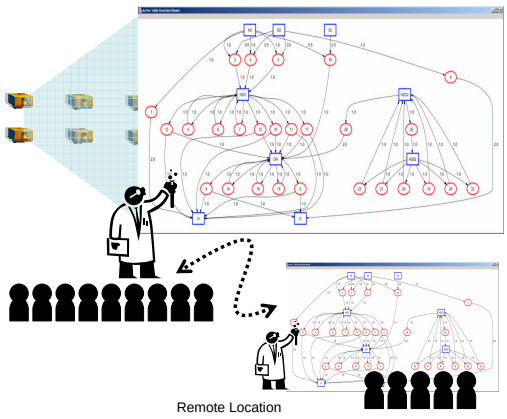

Fig. 6. Cartoon of the usage of an Active Mural in classroom settings.

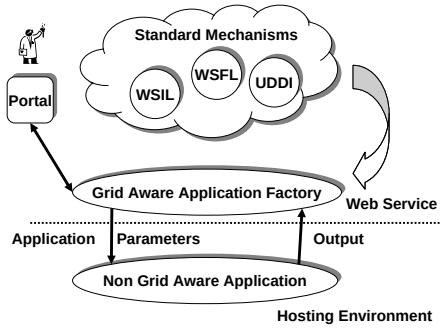

Fig. 7. Exporting a non-Grid-aware application as a Web service

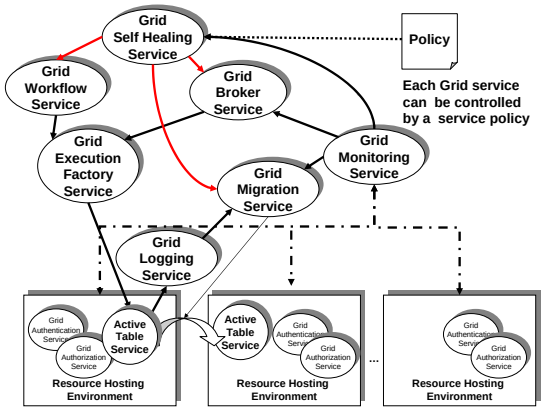

Fig. 8. Advanced Grid services enabling quality-of-service assurances.

11 Conclusion

In this paper we have shown that a service-oriented architecture can be used to provide access to collaborative application environments. We have demonstrated this feature by prototyping an application-oriented portal usable for thermochemical studies. We have developed an architecture that is flexible and open so that additional services can be integrated in our system at a later time. Because of the service-oriented design it will be possible to replace our services in the future with more advanced ones and to customize the behavior of the system with the help of a workflow engine. Because we combine the use of standard commodity technologies with Grid technologies, we are able to create a Grid infrastructure using commodity tools. This initial application has already led to the modification and enhancement of toolkits such as the Java CoG Kit providing access to the Grid.

Acknowledgments

This work was supported by the Division of Chemical Sciences, Geosciences and Biosciences, Office of Basic Energy Sciences, and by the Mathematical, Information, and Computational Science Division subprogram of the Office of Advanced Scientific Computing Research, U.S. Department of Energy, under Contract W-31-109-Eng-38. Funding for the Globus Project research and development is provided by DARPA, DOE, and NSF. The work is performed with help of the SciDAC Collaboratory for Multi-scale Chemical Sciences (CMCS) and the SciDAC Java Commodity Grid project sponsored by the Department of Energy. We thank Jim Meyers, Larry Rahn, David Leahy, David Montana, Karen Schuschardt, Carmen Pancerella, Christine Yang, Ian Foster, the other members of the SciDAC CoG Kit project, the SciDAC CMCS project, and the Globus Project. The Globus Toolkit and Globus Project are trademarks held by the University of Chicago.

References

1. Apache AXIS. http://xml.apache.org/axis/.
2. Condor home page. http://www.cs.wisc.edu/condor/.
3. The Castor Project. http://castor.exolab.org/.
4. The Global Grid Forum Web Page. http://www.gridforum.org.
5. The Jetspeed Webpage. http://jakarta.apache.org/jetspeed/.
6. Universal Description, Discovery and Integration of Business for the Web, June 2002. http://www.uddi.org/.
7. Access Grid. http://www-fp.mcs.anl.gov/fl/accessgrid/default.htm.
8. R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, and V. Welch. A National-Scale Authentication Infrastructure. *IEEE Computer*, 33(12):60–66, 2000.
9. I. Foster, C. Kesselman, and S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of Supercomputer Applications*, 15(3), 2001. http://www.globus.org/research/papers/anatomy.pdf.
10. Ian Foster, Carl Kesselman, Jeffrey Nick, and Steven Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. http://www.globus.org/research/papers/ogsa.pdf, February 2002.

11. M. Frenklach. *Combustion Chemistry*, chapter 7 Modeling, pages 423–453. Springer-Verlag, 1984.
12. GraphViz: A Graph Vizualization Package. AT&T Research, http://www.research.att.com/sw/tools/graphviz/.
13. Grappa, A Java Graph Package. AT&T Research, http://www.research.att.com/~john/Grappa/grappa.html.
14. International Symposium in High Performance and Distributed Computing. *High Performance and Grid Programming in Java and Python*, San Francisco, CA, 6 August 2001. http://www.cogkits.org.
15. E. J. Whitehead Jr. and M. Wiggins. WebDAV: IETF Standard for Collaborative Authoring on the Web. *IEEE Internet Computing*, 2(5):34, September-October 1998.
16. Sriram Krishnan, Patrick Wagstrom, and Gregor von Laszewski. GSFL: A Workflow Framework for Grid Services. In *Preprint ANL/MCS-P980-0802*, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, U.S.A., 2002.
17. Jr. M.W. Chase. NIST-JANAF Thermochemical Tables. Fourth Edition. *Journal of Physical and Chemical Reference*, (Monograph 9), 1998.
18. Carmen M. Pancerella, Larry Rahn, and Christine Yang. The Diesel Combustion Collaboratory: Combustion Researchers Collaborating over the Internet. In *Proceedings of SC99*, Portland, OR, November 13-19 1999. http://www.supercomp.org/sc99/.
19. B. Ruscic, M. Litorja, and R. L. Asher. Ionization Energy of Methylene Revisited: Improved Values for the Enthalpy of Formation of CH2 and the Bond Energy of CH3 via Simultaneous Solution of the Local Thermochemical Network. *J. Phys. Chem. A.*, 103:8625–8633, 1999.
20. B. Ruscic, J. V. Michael, P. C. Redfern, L. A. Curtiss, and K. Raghavachari. Simultaneous Adjustment of Experimentally Based Enthalpies of Formation of CF3X, X = nil, H, Cl, Br, I, CF3, CN, and a Probe of G3 Theory. *J. Phys. Chem. A.*, 102:10889–10899, 1998.
21. SciDAC. http://www.scidac.org/.
22. Gregory P. Smith, David M. Golden, Michael Frenklach, Nigel W. Moriarty, Boris Eiteneer, Mikhail Goldenberg, C. Thomas Bowman, Ronald K. Hanson, Soonho Song, William, C. Gardiner, Jr., Vitali V. Lissianski, and Zhiwei Qin. Gri-Mech: Thermodynamic Data at 298 K, of the Standard Enthalpies of Formation for Radical Species. http://www.me.berkeley.edu/gri_mech/data/thermo_table.html.
23. Gregor von Laszewski and Ian Foster. Grid Infrastructure to Support Science Portals for Large Scale Instruments. In *Proceedings of the Workshop Distributed Computing on the Web (DCW)*, pages 1–16. University of Rostock, Germany, 21-23 June 1999. *(Invited Talk)*.
24. Gregor von Laszewski, Ian Foster, Jarek Gawor, and Peter Lane. A Java Commodity Grid Kit. *Concurrency and Computation: Practice and Experience*, 13(8-9):643–662, 2001.
25. Gregor von Laszewski, Ian Foster, Jarek Gawor, Peter Lane, Nell Rehn, and Mike Russell. Designing Grid-based Problem Solving Environments and Portals. In *34th Hawaiian International Conference on System Science*, Maui, Hawaii, 3-6 January 2001. http://www.mcs.anl.gov/~laszewsk/papers/cog-pse-final.pdf, http://computer.org/Proceedings/hicss/0981/volume%209/0981toc.htm.
26. Gregor von Laszewski, Jarek Gawor, Carlos J. Pe na, and Ian Foster. InfoGram: A Peer-to-Peer Information and Job Submission Service. In *Proceedings of the 11th Symposium on High Performance Distributed Computing*, Edinbrough, U.K., 24-26 July 2002.
27. Gregor von Laszewski, Gail Pieper, and Patrick Wagstrom. *Performance Evaluation and Characterization of Parallel and Distributed Computing Tools*, chapter Gestalt of the Grid. Wiley Book Series on Parallel and Distributed Computing. to be published 2002.
28. Gregor von Laszewski, Kazuyuki Shudo, and Yoichi Muraoka. Grid-based Asynchronous Migration of Execution Context in Java Virtual Machines. In Arndt Bode, Thomas Ludwig, Wolfgang Karl, and Roland Wismüller, editors, *Proceedings of EuroPar 2000*, volume

1900 of *Lecture Notes in Computer Science*, pages 22–34, Munich, Germany, 29 August - 1 September 2000. Springer. *(Invited Talk)*.
29. WebBook: A gateway to the data collections of the National Institute of Standards and Technology. NIST, http://webbook.nist.gov/.