

# A Grid Service Based Active Thermochemical Table Framework

Gregor von Laszewski<sup>2,\*</sup>, Branko Ruscic<sup>1</sup>, Patrick Wagstrom<sup>2</sup>, Sriram Krishnan<sup>2</sup>, Kaizar Amin<sup>2</sup>, Reinhardt Pinzon<sup>1</sup>, Melita L. Morton<sup>1</sup>, Sandra Bittner<sup>2</sup>, Mike Minkoff<sup>2</sup>, Al Wagner<sup>1</sup>, John C. Hewson<sup>3</sup>

<sup>1</sup> Chemistry Division and <sup>2</sup> Mathematics and Computer Science Division  
Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, U.S.A.

<sup>3</sup>Sandia National Laboratories, Livermore, CA 94551-0969, U.S.A.

\* Correspondence: gregor@mcs.anl.gov, Phone: 630 252 0472, Fax: 630 252 1997

## 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 using existing infrastructures and middleware, such as Web Services, the Globus Toolkit [2] 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 which use thermochemical tables. We modified the original approach to constructing thermochemical tables to include an iterative process of refinement leading to increased accuracy, and are in the process of 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 to improve a wide range of scientific questions that lead to better understanding of processes such as climate and combustion [17, 11]. To derive a better understanding of these phenomena it is important to be able to predict and verify them to a high degree of accuracy.

Until now, the thermochemical data necessary to peruse such calculations is unfortunately available only in static table form and the algorithms to derive accurate model descriptions are too imprecise to deal with the complex chemical reactions that we encounter in experiments within state-of-the-art laboratory experiments or processes observed in nature. Our goal is to improve this situation while using innovative algorithms and deliver them to the scientists through an advanced collaborative environment.

Novel modalities of deriving new scientific results can be stimulated while enabling a collaborative environment, in which scientists can publish and share their results with others, perform sophisticated calculations that are otherwise not accessible, and integrate newly developed algorithms as part of a service model in interdisciplinary scientific scenarios.

We observe that the Grid [14, 10, 9] can provide the basic middleware infrastructure for bootstrapping a sophisticated compute end collaborative environment that assists the scientist for the next generation of scientific challenges. Elementary to the Grid is the sharing of resources controlled by different administrative domains. The Grid allows scientists to collaborate even though their resources may be controlled by different domains and their access to these resources is enabled through the use and creation of virtual organizations that allow sharing of resources between different administrative domains.

In this paper, we will show how we provide advanced services that can be accessed collaboratively by the scientists. Their integration as part of a process described as workflow can enable the creation of services that can be easily reused by the community. The scientists are in a

position to concentrate on the science, while application developers will focus their attention to 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 which is directly related to the work we perform. We provide an analysis of a current process to derive thermochemical tables that are 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 will outline our service oriented architecture and outline why services such as security, data transfer, registration, and scheduling assist in assembling such a sophisticated collaborative environment. Before we draw conclusions we point out opportunities for further research. We expect that much of the work presented in this paper will be shown as part of a demo at SC'2002 together with the SciDAC Collaboratory for Multi-scale Chemical Sciences (CMCS) and the SciDAC Java Commodity Grid project sponsored by the Department of Energy.

## 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* ( $\Delta H_f^\circ$ ), 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 as shown in Figure 1.

In this diagram the equations can be expressed as a graph that contains horizontal lines representing different values of the enthalpy. Typically the differences between these values are determined experimentally or can be derived using the thermodynamic laws from other enthalpy values. However, values obtained from experiments 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 natural to visualize the transition with directed edges between the states. An alternative graph is displayed on the upper right

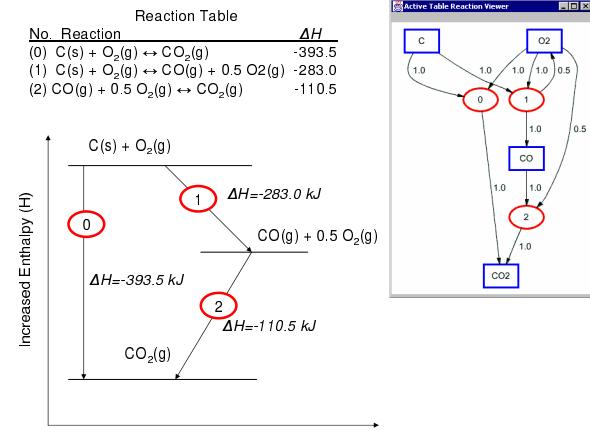


Figure 1: Enthalpy diagrams and thermochemical reaction tables.

hand corner that emphasizes the possible states in which a chemical species (e.g., 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 ( $\Delta H_f^\circ$ ) are developed with the help of an elaborate *sequential process*. In each step a new species is added while all  $\Delta H_f^\circ$  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:  $C_p^\circ, S^\circ, (H_T^\circ - H_0^\circ), \Delta H_{fT}^\circ, \Delta G_T^\circ$ , are developed by determining the partition function  $Q, \ln Q, T\partial(\ln Q)/\partial T$ , and  $T^2\partial^2(\ln Q)/\partial T^2$  from the species-specified quantities and the newly selected single temperature enthalpy. (For more information about the aforementioned terms, please refer to [19].)

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

This sequential approach has several disadvantages and results in  $\Delta H_f^\circ$  and their 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 rela-

tionships in conventional tables produce thermodynamic tables that are static in its nature. The proper update with new knowledge is nearly impossible since forward relationships are non-transparent and updating one species may improve things locally, but increases the global inconsistencies.

A new approach is needed that circumvents these disadvantages.

### 3 Active Thermochemical Tables

The most important difference is in how the species-interdependent data is treated. The Active Table approaches the information content hidden in the species-interdependent data from the viewpoint of a thermochemical network [19, 18]. 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 over-determined. 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 both in 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  $\chi^2$  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, and once the optimal solution to the network has been found, to calculate the partition functions and hence develop the temperature dependence of the standard thermochemical functions. It should be noted that, as opposed to the simultaneous solution described above, the traditional sequential approach would correspond to a procedure where 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 over-determined 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/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 is 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, via 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 test corresponds to a new experiment, provides feedback on sensitivity of the network to various measurements. The approach has also potential to become an interesting learning and education tool, etc. An Active Table can provide a ranked list of links that are missing or weak from a statistical viewpoint, i.e. it can provide pointers to what would be the most useful new experiments or calculations.

### 5 Application Requirements

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

- the calculation and the visualization of an active thermochemical table,
- 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,

- the query of data needed for the polynomial fitting and the Active Tables.

In the next section we will extend our requirement list beyond the pure scientific application need and concentrate more on the modalities of using these applications in a shared and collaborative environment.

## 6 Grid Requirements

To decide upon a computational environment that supports our algorithmic proposed solution we first have to analyze a set of requirements that are imposed on us based on the modality of the scientific research to be performed. We identify that we have a need for a

**Collaborative** environment that supports the interaction amongst 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 straight forward fashion.

**Adaptive** environment that is able to be flexible towards future changes based on hardware and software.

**Dynamic** environment that allows the creation of transient services to enable ad-hoc collaboration and usage of other application services.

These requirements are in common with many other scientific disciplines and a large amount of research has been performed in the last 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 to utilize the anticipated availability of standardized Grids as defined by the Grid Forum.

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. We draw upon this rich experience and prototype an architecture that can be supported by these commodity integrating technologies.

A further analysis of our problem domain revealed that it is beneficial to build the framework based on a service oriented architecture. The 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 sets of services:

**Grid Broker Service** to deal with large number of calculations that are involved with future large scale reactions and their real time requirement for allowing interactive use [22, 1].

**Grid Workflow Service** to enable the interplay of Grid services through workflow descriptions [23, 24].

**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) [23, 26].

**Grid Monitoring Service** to monitor the state of the hosting environment so that feedback to Grid services is provided the need to react on state changes to the services [25].

**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 according to performance and quality of service descriptions and policies [27].

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

**Grid Self Healing Service** that determines a policy of how and when it is necessary to change the dynamically instantiated Grid workflow applications. This includes preventive measurements such as, service replication, service migration, service check pointing, and service monitoring.

**Collaborative Steering Service** is needed to collaboratively work on the creation of data, thoughts, and ideas that will lead to new scientific findings.

**Portal Service** that simplifies the interaction with this sophisticated environment, since the scientist should concentrate on the science and not the environment [14].

Although many more services are needed, we decided to restrict our initial prototype on these sets of services, which we will either improve or prototype.

## 7 Architecture

Based on our previous discussion 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 have depicted these application specific services in Figure 2.

The scientist has the ability to 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 [16, 28, 21], the query of thermochemical data for species, the creation of a database base on species data, and the calculation of an Active Table based on a number of reactions (currently under development), as well as 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 on a geographically dispersed environment. Thus it will be possible to maintain changes to the original algorithm by the application specialist, while at the same time minimizing the effort for deployment 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, making it effectively able to expose a customized functionality to disparate user communities.

In our first prototype we provide a Swing portal that we will integrate into portlets exposed through a JetSpeed portal framework.

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

**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  $\chi^2$  minimization of the systems of

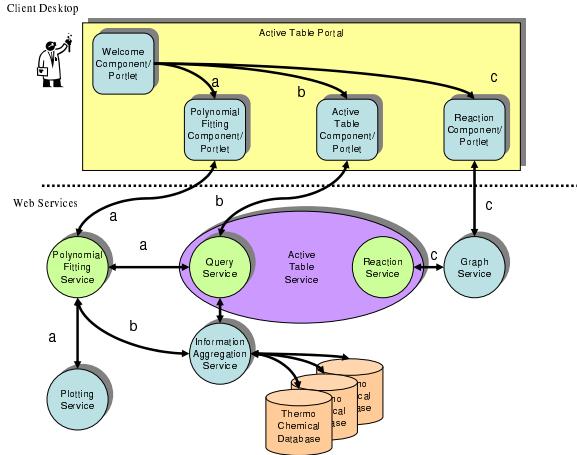


Figure 2: The application specific services to enable the scientist to access a convenient portal interfacing exposing active thermochemical tables, polynomial fitting of JANAF data, and their visualization.

linear equations obtained from a reaction table. This service is composed of :

**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 which is currently using the GraphViz Engine [12, 13].

**Information Aggregation Service** that will allow querying multiple databases maintained in a distributed fashion. A caching mechanism will minimize the search latency for frequently asked queries.

## 8 Example Use Cases

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

**Use Case A: Requesting a Polynomial Approximation of the thermochemical characteristics of Argon.** A user wishing to perform this task would start up the poly-fit client on their desktop and request to receive a mathematical approximation for the thermochemistry properties of a given substance, for example Argon. The poly-fit client would then open a communication channel to a

polyfit web service to send the request. In order for the polyfit service to process the request, it needs the basic thermo data for Argon, which can be 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 then the user is able to view the data on his/her terminal. The workflow for the components and services that are involved in this use case are augmented with the letter *a* in Figure 2. The portal interface is shown in Figure 3.

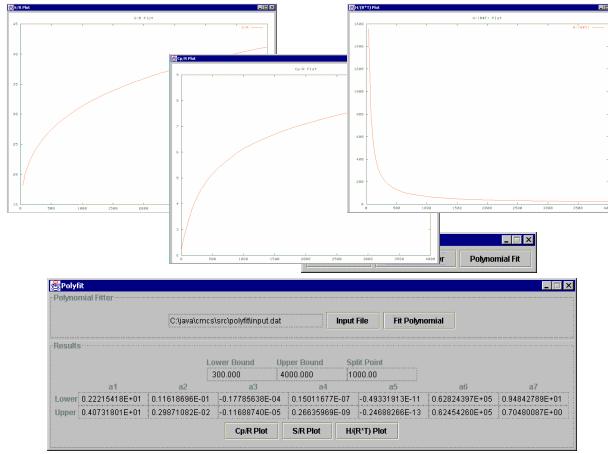


Figure 3: The polynomial fitting of data based on JANAF tables

**Use Case B: Visualization of a reaction graph and obtaining information on the species in the graph.** A user wishing to perform visualization would first start up the reaction visualization client. They would then submit their reaction file to this client. A reaction file could be as simple as one chemical reaction, or it could be hundreds of reactions all working together. The client would then pass 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 augmented with the letter *b* in Figure 2. The portal interface is shown in Figure 4 with the reactions as depicted in Table 1.

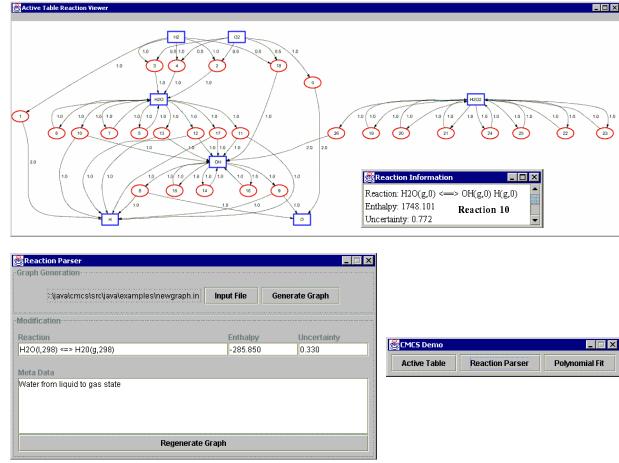


Figure 4: A reaction graph visualization component.

Table 1: A thermochemical reaction table

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

**Use Case C: Requesting the thermo table for carbon (graphite).** Initially, the user would start up an Active Tables client on their computer. The user would then enter their 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 enough information to process the request, then it may request more information from a WebDAV [8] server, or other web service. That new information is returned to the Active Tables program. This is then passed back to the Active Tables 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 augmented with the letters *a* and *c* in Figure 2. The portal interface is shown in Figure 5

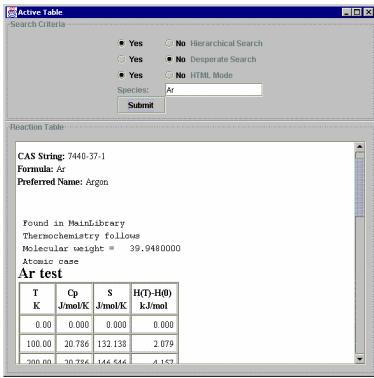


Figure 5: Querying the species dictionary

**Use Case D (currently in planning): Using the Active Table in educational outreach projects** Because we use standard Grid security infrastructure [7] and the ability of flexibly assembling our services it will be possible to create customized portal access through the *Access Grid* [6] that will allow us to share and display our interface through active murals amongst a set of participating institutions in an educational setting. The ongoing DOE sponsored SciDAC projects [20] 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.

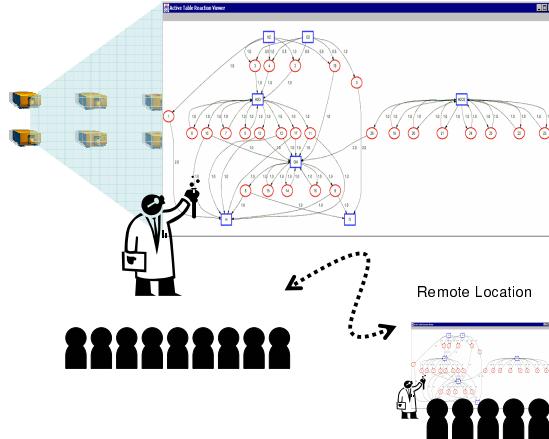


Figure 6: Cartoon of the usage of an active Mural in class room settings.

## 9 Current and Future Research

Currently, we have a usable prototypical implementation, but we foresee enhancements to our framework on mul-

tiple levels. 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. The service is also capable of returning the standard output and error streams of the executable to the caller, apart from being able to return the basic exit value of the executable. We use such a service to export as a web service Fortran codes written by the chemists, without having to modify them. Clients can retrieve a handle to such a service using standard mechanisms (e.g UDDI [5]), and access the scientific codes, as shown in Figure 7. We plan to write a generic tool that will be able to expose any non-grid aware code, as a grid service. This tool will accept 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 generate the required glue code to expose it as a service. The implementation will be along the lines of the WSDL to Java convertor tool provided by Axis [3], and in fact, will involve modification of the code generation provided by Axis to suit our needs.

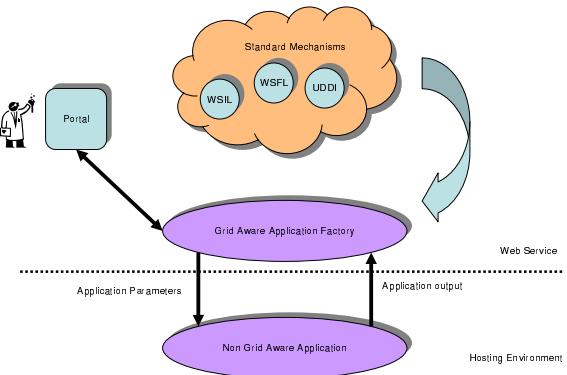


Figure 7: Exporting a non-grid aware application as a Web Service

The graph service is also exported as a web service, for the same reasons. The reactions which are fed into the graph service in a format convenient to the chemists, are then parsed and converted to a canonical XML format, using Castor [4], which is a data binding framework for Java. The standard XML representation with corresponding Java bindings was a logical choice, as we could 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, and it is anticipated that we will have several ways to visualize

these graphs. The XML format also lends itself very well to being transferred around, and can be visualized locally if need be, to save bandwidth.

In the future, 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, 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 on how we will use these services (currently under development) that allow for reliable and flexible brokering of jobs in Grids with quality of service assurances.

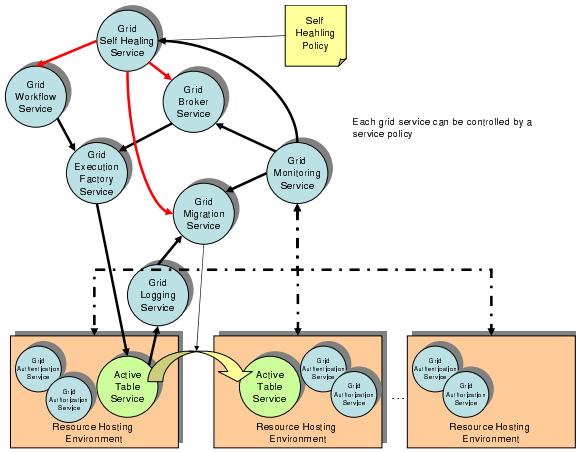


Figure 8: Advanced Grid services enabling quality of service assurances.

One of the requirements that is desirable is the persistent availability of the service to the application user. The easiest way to achieve this is to deploy all services on the users machine. However, this is often not possible and prevents the collaborative access to dynamically changing information contributed by teams of scientists. Hence, it is worthwhile to investigate the design of a high available service, that can adapt itself to the disruptive nature of the Grid and the Internet. To design such a service we plan to use a Grid Monitoring service to observe the state and the condition of the hosting environment in which jobs are executed with the help of brokering services and execution factories. In case dependencies between tasks/jobs need to be expressed, the Grid Workflow Service can be utilized. Upon predetermined conditions and quality of service assurances, tasks/jobs may migrate to other hosting environments to finish their computations. Job migration can be provided by a logging service, that allows the checkpointing of the application and the restart at another location. Currently, we are designing a set of classes to

enable self healing applications.

## 10 Conclusion

In this paper have shown that a service oriented architecture can be used to provide access to collaboratively used application environments. We have demonstrated this feature while 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. Due to its services oriented design it will be possible to replace our services in the future with more advanced ones, and also to customize the behavior of the system with the help of a workflow engine. As we combine the use of standard commodity technologies with Grid technologies, we are able to create a secure Grid infrastructure, using commodity tools. This initial application has already lead to the modification and enhancement of toolkits such as the Java CoG Kit toolkit providing us access to the Grid.

## Acknowledgments

This work was supported 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. DARPA, DOE, NSF support for the Globus Toolkit research and development, and the Division of Chemical Sciences, Geosciences and Biosciences, Office of Basic Energy Sciences. We thank Jim Meyers, Larry Rahn, David Lehaey, David Montana, Karen Schuschardt, Carmen Pancerella, Christine Yang, Ian Foster, Jarek Gawor, and the other members of the SciDAC CoG Kit, SciDAC CMCS projects, and the Globus Project for the valuable discussions leading up to this work. The Globus Toolkit and Globus Project are trademarks held by the University of Chicago.

## References

- [1] Condor Home Page. <http://www.cs.wisc.edu/condor/>, Feb 2002.
- [2] The Globus Project Homepage. <http://www.globus.org/>, April 2002.
- [3] Apache AXIS, June 2002. <http://xml.apache.org/axis/>.
- [4] The Castor Project, June 2002. <http://castor.exolab.org/>.

- [5] Universal Description, Discovery and Integration of Business for the Web, June 2002. <http://www.uddi.org/>.
- [6] Access Grid. <http://www-fp.mcs.anl.gov/fl/accessgrid/default.htm>.
- [7] BUTLER, R., ENGERT, D., FOSTER, I., KESSELMAN, C., TUECKE, S., VOLMER, J., AND WELCH, V. A National-Scale Authentication Infrastructure. *IEEE Computer* 33, 12 (2000), 60–66.
- [8] E.J. WHITEHEAD, J., AND WIGGINS, M. WebDAV: IETF Standard for Collaborative Authoring on the Web. *IEEE Internet Computing* 2, 5 (September–October 1998), pp. 34.
- [9] FOSTER, I., KESSELMAN, C., NICK, J., AND TUECKE, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. <http://www.globus.org/research/papers/ogsa.pdf>, February 2002.
- [10] FOSTER, I., KESSELMAN, C., AND TUECKE, S. 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>.
- [11] FRENKLACH, M. *Combustion Chemistry*. Springer-Verlag, 1984, ch. Modeling, Chap. 7, pp. 423–453.
- [12] GraphViz, A Graph visualization 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] HARIRI, S., AND PARASHAR, M., Eds. *Performance Evaluation and Characterization of Parallel and Distributed Computing Tools*. Wiley Book Series on Parallel and Distributed Computing. to be published, ch. Gestalt of the Grid.
- [15] KENDALL, R. A., APR, E., BERNHOLDT, D. E., BYLASKA, E. J., DUPUIS, M., FANN, G. I., HARRISON, R. J., JU, J., NICHOLS, J. A., NIEPLOCHA, J., STRAATSMA, T. P., WINDUS, T. L., AND WONG, A. T. High Performance Computational Chemistry; an Overview of NWChem a Distributed Parallel Application. *Computer Physics Communications* 128 (2000), pp. 260.
- [16] M.W. CHASE, J. NIST-JANAF Thermochemical Tables. Fourth Edition. *Journal of Physical and Chemical Reference*, Monograph 9 (1998).
- [17] PANCERELLA, C. M., RAHN, L., AND YANG, C. 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/>.
- [18] RUSCIC, B., LITORJA, M., AND ASHER, R. L. Ionization Energy of Methylene Revisited: Improved Values for the Enthalpy of Formation of CH<sub>2</sub> and the Bond Energy of CH<sub>3</sub> via Simultaneous Solution of the Local Thermochemical Network. *J. Phys. Chem. A.* 103 (1999), 8625–8633.
- [19] RUSCIC, B., MICHAEL, J. V., REDFERN, P. C., CURTISS, L. A., AND RAGHAVACHARI, K. Simultaneous Adjustment of Experimentally Based Enthalpies of Formation of CF<sub>3</sub>X, X = nil, H, Cl, Br, I, CF<sub>3</sub>, CN, and a Probe of G3 Theory. *J. Phys. Chem. A.* 102 (1998), 10889–10899.
- [20] SciDAC. <http://www.scidac.org/>.
- [21] SMITH, G. P., GOLDEN, D. M., FRENKLACH, M., MORIARTY, N. W., EITENEER, B., GOLDENBERG, M., BOWMAN, C. T., HANSON, R. K., SONG, S., WILLIAM, GARDINER, JR., C., LISSIANSKI, V. V., AND QIN, Z. 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](http://www.me.berkeley.edu/gri_mech/data/thermo_table.html).
- [22] von LASZEWSKI, G., AND FOSTER, I. Grid Infrastructure to Support Science Portals for Large Scale Instruments. In *Proc. of the Workshop Distributed Computing on the Web (DCW)* (June 1999), University of Rostock, Germany.
- [23] von LASZEWSKI, G., FOSTER, I., GAWOR, J., AND LANE, P. A Java Commodity Grid Kit. *Concurrency and Computation: Practice and Experience* 13, 8–9 (2001), 643–662. <http://www.globus.org/cog/documentation/papers/cog-cpe-final.pdf>.
- [24] von LASZEWSKI, G., FOSTER, I., GAWOR, J., LANE, P., REHN, N., AND RUSSELL, M. Designing Grid-based Problem Solving Environments and Portals. In *34th Hawaiian International Conference on System Science* (Maui, Hawaii, January 3–6 2001). <http://www.mcs.anl.gov/~laszewsk/papers/cog-pse-final.pdf>.
- [25] von LASZEWSKI, G., GAWOR, J., NA, C. J. P., AND FOSTER, I. InfoGram: A peer-to-peer information and job submission service. In *Proceedings*

- of the 11th Symposium on High Performance Distributed Computing (HPDC-11)* (Edinburgh, Scotland, July 24-26 2002). <http://www.globus.org/cog>.
- [26] VON LASZEWSKI, G., AND JACKSON, K., Eds. *Using Java and Python in Grids* (2001). <http://www.cogkits.org>.
- [27] VON LASZEWSKI, G., SHUDO, K., AND MURAKA, Y. Grid-based Asynchronous Migration of Execution Context in Java Virtual Machines. In *Proceedings of EuroPar 2000* (Munich, Germany, 2000), Lecture Notes in Computer Science, Springer.
- [28] WebBook: A gateway to the data collections of the National Institute of Standards and Technology. NIST, <http://webbook.nist.gov/>.