

Management of Web and Associated Grid Technologies for Quantum Chemistry Computation

Kim K. Baldridge¹ and Jerry P. Greenberg¹

¹San Diego Supercomputer Center
9500 Gilman Drive
La Jolla, CA 92037-0505
{kimb, jpg}@sdsc.edu

Abstract. We present a description of our ongoing efforts to utilize the services provided over the World Wide Web to advance the field of computational quantum chemistry utilized for detailed study of the electronic structure and properties of molecular systems, as well as the dissemination of results both for research and education. There are several facets of these efforts: (1) development of a computational chemistry web portal (2) development of an XML schema based on output data of electronic structure software, and (3) development of database and associated query tools that will serve as a basis for storage, retrieval, and manipulation of QM data in uniquely new ways. These efforts are distinctly coupled together through our creation of a “workflow” of web services, using grid resources.

1 Introduction

Advances in computational science technologies continually present researchers with new opportunities for solving problems. However, at the same time, such technologies often present a bewildering set of choices for computing paradigms, and additionally can result in overwhelming amounts of data to process. While computational resources have grown in power and size, they have associated complexity issues to negotiate. Thus, while one can now obtain unprecedented speed, memory and storage capacities, there is an associated plethora of platforms, with different operating systems, queuing mechanisms, directory structures, and access methods. We have recently described the use of web service technology to facilitate quantum chemistry (QC) calculations, shielding the user from some of these complexities[1]. Here, we expand on those efforts with an emphasis on the design of XML schema.[2]

A parallel effort in the development of QM database resources is intended to store, in a systematic manner, data obtained from QC calculations. Such a database can serve several purposes, including, a) providing a source for querying chemical properties over a range of chemical compounds, b) comparing new results with previous calculations, or levels of theory, or c) accessing accurate data for chemical analysis, or for use in one of a variety of empirical models. Such inquiries are somewhat difficult now since results of QC calculations are often stored as “flat” files that are human readable only, or at best, parsed by shell scripts. Comparisons become difficult not only because of different storage formats, but also because of

accessibility: researchers may not know about previous calculations, and it may not be clear how to compare data.

The process of building a QM-DB and working with data mining and query experts, enables one to not just build a database of chemical quantities, but provides a tool that accommodates extensive manipulation of the quantum data. Specifically, access to of query capabilities on top of a rigorous QM database offers the ability to query data in a ways that were unavailable, or, even more so, not thought of due to the inaccessibility – one can now imagine cross coupling data types that can answer deeper chemical questions than previously enabled, or construction of creative theoretical methods that span molecular scale.

2 The GAMESS Portal

Our first effort to utilize web services involved the creation of a GAMESS (General Atomic and Molecular Structure System)[3] portal (gridport.npaci.edu/games). GAMESS is an *ab initio* electronic structure code in which the standard release reads and writes conventional input and output files. The most basic input file (.inp) consists of atomic coordinate data and associated symmetry information, basis set specification, and keywords describing the type of calculation(s) to perform. The output consists of files containing virtually all user information pertaining to the actual computational process and associated results (.out), as well as a data summary file (.dat) that may be used for data analysis, visualization and/or for starting new runs based on the data from the old. The initial motivation for the GAMESS portal was to isolate users from the complexities of the computational grid by providing a standard user interface for accessing input and output, running jobs on one of a variety of platforms without logging onto those platforms, and the ability to transfer data among various platforms.

The portal has facilities for processing the output of a particular run via visualization with our computational chemistry visualization and analysis tool, QMView. However, the major focus to date has been facilitation of the processes of computing and storing data. The existing portal enables facile job submission to remote platforms, and retrieval of results via any commodity browser. More recently, storage for input and output data has been enabled through connections to the SDSC Storage Research Broker (SRB). [4] The SRB additionally provides the abstraction mechanisms needed to implement data grids, digital libraries, and persistent archives for data sharing, data publication, and data preservation. Thus, things like the quantum mechanical database project as well as other planned digital libraries of QM data and images, can easily involve more sites (institutions) as the library grows. The SRB will register digital entities of any format into a logical name space to create global, persistent identifiers in the digital library. The SRB is currently used in production or being evaluated across multiple projects at NSF, NASA, DOE, DOD, NIH, NLM, NARA, and the Library of Congress. Our existing portal invokes the persistent archiving capability of the SRB to store its digital entities from the GAMESS portal.

The underlying basis for the chemistry portals is the Gridport Tool Kit [5] which in turn provides an interface to the GLOBUS toolkit [6] for secure transfer of files and execution of remote procedure calls to any high performance computing (HPC) resource. The GAMESS portal submission page is illustrated in Figure 1.

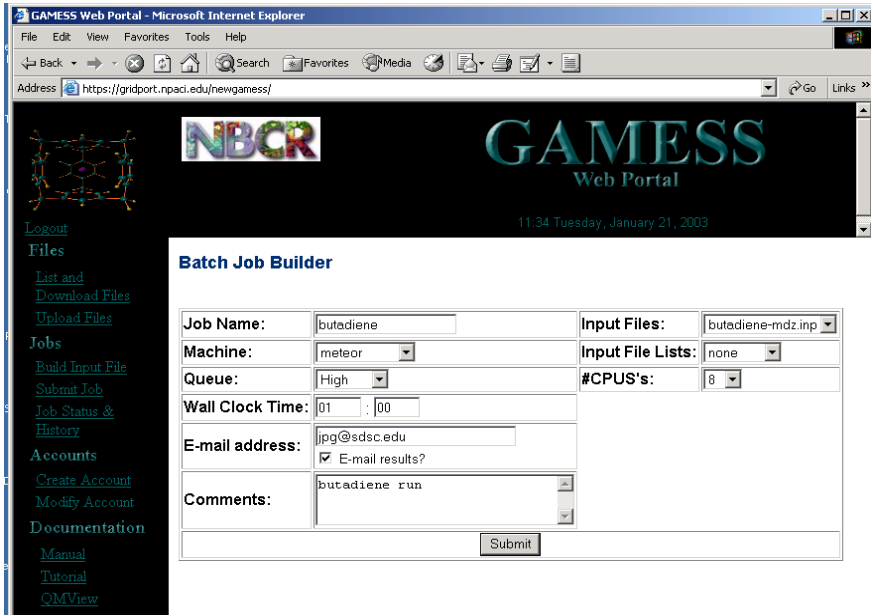


Fig 1. The submit page from the GAMESS PORTAL

One can choose an input file from a list of previously uploaded files, select the remote resource to run on, and choose from such options as the queue name, number of processors, and maximum time. The portal informs the user of job status and enables results to be stored in the SRB in a specific user data collection space.

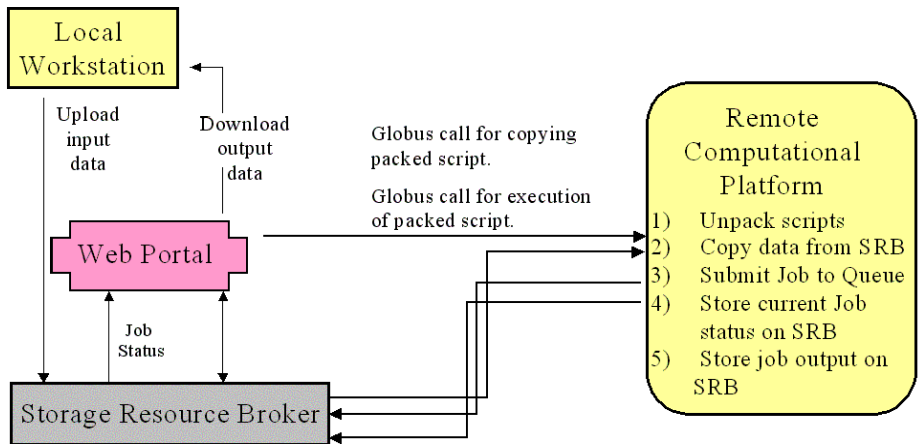


Fig. 2. Flow chart illustrating the operation of the GAMESS portal.

Because of the general need for secure communication (authenticated and perhaps confidential communication) between the elements of a computational Grid, we have incorporated Grid Security Infrastructure (GSI) [7]. Other major incentives for including such security within the portal include the need to support security across organizational boundaries (thereby eliminating the need for a centrally-managed security system), and the need to support “single sign-on” for users of the Grid, including designation of authorization for computations that involve multiple resources. Thus, for example, authentication for retrieving and copying files from a remote platform to the SRB is accomplished using this technology, as illustrated in Figure 2.

The design of the SDSC science portals has undergone several renovations as we leverage what we have learned in the construction of the GAMESS portal in order to “portalize” other computational chemistry and biochemistry software. One of the most notable extensions came with collaborative work in conjunction with Pevzner and his group at the Bioinformatics Lab at UCSD. Their work involved the development of the “EULER” [8] software, a new technique for genetics sequencing. The construction of a portal for their application enabled important enhancements to our existing portal in general, as a result of the specialized needs of extending this application as a service to a general community on a few nodes of a commodity cluster. Portal enhancements included a) the optimization of high throughput for job submissions, b) improved formatting and presentation of the output, and c) enhanced facilities for downloading results.

Portal optimization for high throughput job submissions was accomplished primarily by reducing the number of remote procedure calls to the remote resource. Such calls were reduced to one call to copy a shell script and one call to execute the script. The shell script “unpacks” several additional scripts, places them in the appropriate directory, and then executes them appropriately. The unpacked scripts retrieve input data from the SRB and submit the job to the queue. In order to eliminate calls to the remote machine that inquire about job status (e.g., queued, running, or finished status), the running batch script sends a file containing the current job status directly to the SRB. The web portal then copies and reads this file in order to report job status. The resulting procedure is much less costly in terms of time.

An example of the portal output data page is given in Figure 3. QMView (see below) may be registered as a recognized MIME type in order to view output files by clicking on them in the browser. In addition, there are facilities for running other programs with the selected files as input. Files may also be viewed within the browser or, one or more files may be selected for downloading in one of several archiving formats. In addition, an error reporting facility has been added to send examples of bugs to the portal developers. Users may submit reports of bugs along with all relevant files.

In order to facilitate access to clusters by users who may not have an account on those systems, we have taken advantage of the sub-account system within the gridport toolkit in order to allow users who have a portal account to run jobs on a cluster under a master account. The “owners” of the portal may add sub-accounts from the portal master account. The mapping of the portal accounts to the master account on the remote cluster is handled by Globus. This has been implemented within the Euler portal, where users may now run jobs on a small dedicated cluster. We intend to extend these service-oriented capabilities to other applications, including GAMESS.

The screenshot shows the GAMESS Web Portal interface. The browser window title is "GAMESS Web Portal" and the address bar shows "https://gridport.ncaci.edu/newgameSS/". The page features a navigation menu on the left with links for Logout, Files, Job, Accounts, and Documentation. The main content area displays a "GAMESS File Listing" table with columns for Job, Viewer, Size, and Date. Below the table are options to "Select" or "unselect" all files, and an "Action to perform on selected files:" section with buttons for Download, Submit files, and Delete.

Job	Viewer	Size	Date
Up to Completed Jobs			
butadiene-mdz.dat	Text	722.5K	21 Jan 2003 12:38
butadiene-mdz.inp	Text	514	21 Jan 2003 12:38
butadiene-mdz.log	Text	349.6K	21 Jan 2003 12:38
butadiene_butadiene-mdz.batch16	Text	5.3K	21 Jan 2003 12:38
butadiene_butadiene-mdz.params	Text	244	21 Jan 2003 12:38
butadiene_butadiene-mdz.remotedir	Text	60	21 Jan 2003 12:38
butadiene_butadiene-mdz.sp2hosts	Text	265	21 Jan 2003 12:38
butadiene_butadiene-mdz.run.csh	Text	1.0K	21 Jan 2003 12:38
butadiene_butadiene-mdz.run2.csh	Text	790	21 Jan 2003 12:38
butadiene_butadiene-mdz.run3.csh	Text		21 Jan 2003 12:38
butadiene_butadiene-mdz.run4.csh	Text		21 Jan 2003 12:38
param.temp	Text	242	21 Jan 2003 12:38
srbout	Text		21 Jan 2003 12:38

Fig. 3. The GAMESS Portal Output Page

2.1 Visualization and Analysis Extensions to the Portal

The discussion so far has centered on the mechanism of running a job and has not dealt at all with handling and processing of the data. For these latter capabilities, we have made initial efforts to produce structured output documents directly from the *ab initio* software, GAMESS. As previously mentioned, there are many advantages of structured data over “flat files” of data. The most commonly used format for accessing such data is via Extensible Markup Language, XML. XML is rapidly becoming the industry standard for transferring data between web services [9] and actually facilitates not only data analysis, but also transfer, storage and retrieval.

There has been previous work in creating schemas specifically for chemical data such as developed in Chemical Markup Language, CML [10]. We have created our own schema whose focus is on quantum chemistry computational data but could in principle be modified and incorporated into the CML schema using the CML data elements. For now, the data element names and structure resemble the output of a GAMESS job. That is, at least at the initial stages, the elements and attributes are specifically related to *ab initio* quantum mechanical calculations, and to GAMESS in particular. This schema is being implemented directly within the GAMESS software, and additionally within our visualization and analysis program, QMView, which is described next. The goal is to have a workflow of XML documents for a variety of functional purposes, including the generation of input and output and the deposit of output into a database that will subsequently be available for querying.

Currently, the graphical interfaces that are available for *ab initio* electronic structure software are limited in terms of computational flexibility and in graphics output capabilities. We have already written packages including QMView, (Quantum

Mechanical Viewing tool) [11],[12] and MDMovie (Molecular Dynamics Movie [13], that have addressed these needs, in the former case for quantum chemistry, and the latter, for molecular dynamics. The software is available under the OpenGLTM graphics library, which allows our interactive 3-D software to run on many types of workstations and personal computers.

It is the nature of QM calculations that a relatively small amount of data is required for input to the software, but large volumes of data (e.g., 3-dimensional molecular orbital, vibrational, electrostatic, and dynamical data) are output. Our initial efforts in the development of QMView were devoted to the visualization of calculated quantum mechanical data. The continued development of visualization now enables one to generate images for purposes that range from publication, to creation of illustrative animations of, say, reaction pathways, dynamics and vibrational phenomenon. Analysis capabilities encompass a broad range of manipulation of structure, property and differencing techniques. Modifications and revisions are made continually based on our own scientific needs, as well as those of the community of users (Figure 4).

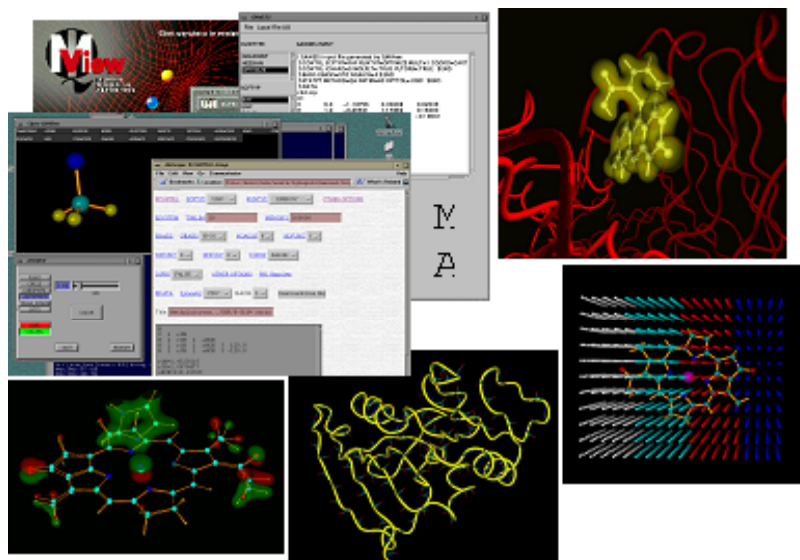


Fig. 4. A Sample of QMView interface and output type.

More recently, QMView is evolving into a distributed computational chemistry environment for building, simulating, and analyzing chemical structures, as well as a central controller for access to other resources (e.g., databases, auxiliary programs etc). In the context of the GAMESS portal, and as a part of our overall project of incorporating grid technologies into quantum chemistry applications, QMView is developing into a front-end for accessing several services including, for example:

- a) Protein DataBank (PDB): Extracting the coordinate data of biopolymers and ligands from the PDB, (e.g, GAMESS input, display of volumetric data . [14]

- b) Compute Engines (e.g., GAMESS, G98, MOPAC): Calculation of coordinate and volumetric data, and subsequent visualization and analysis.
- c) QM-Database (QM-DB): Extraction of data from QM-DB, followed by analysis, correlation, or computation of new fragments and deposit back into the QM-DB.
- d) WebViewing: Exportation of XML files of chemical data from QMView for DB storage, manipulation, and/or viewing within a browser.
- e) Differencing Tools: Analysis of volumetric data, including a variety of differencing schemes and capabilities.
- f) Grid-Workflow Environment: Serves as a full working discovery environment for computational chemistry research, with access to a full set of resources.

2.2 Portal Technology Transfer

GAMESS was the first computational science web portal built using the SDSC GridPort Toolkit (<http://gridport.sdsc.edu/>). The software development package consists of standard, portable technologies with which developers can quickly create Web interfaces to scientific applications. Such application portals can then be used to securely access high-end computing resources from any Web-connected device, including wireless handheld devices. Any application portal built on this toolkit automatically inherits these "HPC anywhere" capabilities. Therefore, we have used what we have learned in developing the GAMESS portal to facilitate other applications. Typically we have found that enhancements or needs of any particular application are quite easily propagated to a new portal environment from the GAMESS portal base.

We now have a variety of useable computational (bio)chemistry and bioinformatics portals that we are offering to the community through SDSC and the National Computational Biomedical Resource. In addition to GAMESS and EULER, these include the APBS (Adaptive Poisson-Boltzmann Solver) [15] portal for calculation of electrostatic potential surfaces for molecular constructs, CE (Combinatorial Extension) [16] portal for performing structural comparison of proteins, Cardiac Modeling [17] portal for electrocardiogram modeling of the heart, and the LAPK portal for Population Pharmacokinetics [18].

While these types of portals have lifted considerable complexity from the attention of the user, it is clear that once this basic portal technology is enabled for an application, one begins to envision more functionality that would further enhance science capabilities. In particular, interoperability within an application, between one application and another, or even between one or more applications and other resources would be the next goal. Thus, in a more general instantiation of these technologies, we are in the process of designing "workflow environments" that will enable more advanced working environments for scientific discovery. These designs, as was true for the portal designs, will be suitable for a wide variety of scientific applications. These efforts are outlined elsewhere [1], and so here we will concentrate on the general relevance and applicability of web services, and the use of the computational grid for the GAMESS software.

3 Advancements towards Work-Flow and Data Sharing

The most common way to extract data from output files is by processing them with command language scripts. Extraction of data from output files using scripts eliminates the need to incorporate the data extraction program into the simulation program. However there are significant disadvantages to the use of common language scripts. A small change in output format can result in the script to fail or a misinterpretation of data. Significant changes in output format may require rewriting the entire script. By placing structured data output facilities within the program, the effort and code expended on data parsing is removed, and the software becomes independent of the format of the “human readable” output file.

As a step in this direction, we initially embedded calls within the FORTRAN routines of the GAMESS program, which prints out an XML document according to a Document Type Definition, DTD. More recently, we have devised an a schema for this purpose.

An effort has been made to put the minimal amount of code into the GAMESS FORTRAN modules for this purpose. The code that outputs the XML document is embedded in a C library that is easily linked to the GAMESS executable. For example, the code fragment below shows how the atomic coordinates are written to the XML document:

```
call output_coords_init(UNITS,4,AORIG,8,IXCOUNT)
call energyxml(ETOT,EUNITS,ILEN,IXCOUNT)
DO 310 IAT=1,NAT
  IF(SKIP(IAT)) GO TO 310
  ZNUC = ZAN(IAT) + IZCORE(IAT)
  X = UNIT * C(1,IAT)
  Y = UNIT * C(2,IAT)
  Z = UNIT * C(3,IAT)
C   IF (MASWRK) WRITE(LU,9030) ANAM(IAT),BNAM(IAT),ZNUC,X,Y,Z
  IF (MASWRK) call output_coord(ANAM(IAT),BNAM(IAT),ILEN,ZNUC
1      ,X,Y,Z)
310 CONTINUE
```

The call to “output_coords_init” sets up the “SYSTEM_STATE” element, which we refer to again below, while the call to “energyxml” prints out the energy. The loop is over all atoms in the molecule and prints the “ATOM_POSITION” elements. The coordinates and energy are stored in common blocks that already exist in the GAMESS program. For run types that generate multiple coordinate sets (such as an Internal Reaction Coordinate run, or a molecular optimization run), the address of the starting point of the molecular orbital data is passed to a C routine. In the case of a run with no coordinate updates, the molecular orbital coefficients are stored in dynamic memory allocated by the C XML library, and then printed out when the coordinates are printed.

3.1 Schema

The schema in its initial implementation defines a document for storing only basic data, such as atomic coordinates, atom types, energies and molecular orbitals. In principle, this is enough data to evaluate further properties of a molecular system. However, in practice and for convenience, we plan to add other system properties such as gradient and hessian information, solvation information, and other such data, that could as well be evaluated from the essential data.

The data that characterizes a particular run is initially specified by including the job type (RUNTYP), wavefunction type (SCFTYP), the basis set (BASIS), and symmetry information. All choices for run specification type (e.g., energy, optimization, hessian, etc), wavefunction type (e.g., closed shell, RHF, open shell, UHF, etc), basis sets (e.g., 6-31G(d,p), cc-pVDZ, etc), and symmetry (CNV, DND, etc.) are enumerated. Where possible, error checking is being incorporated into the schema. For example, limits specifying the correct choices for parameters such as NGAUSS (number of Gaussians per atomic orbital), correct numeric values for NDFUNC (number of 'd' type polarization functions for heavy atoms), and/or job type combinations can be monitored.

The output data follows with the coordinate data along with all properties associated with that data (e.g., energy, etc.), grouped into an "system_state" set. These properties include even the molecular orbital coefficients for that coordinate set, and in the future will include the vibrational mode vectors. In the case of several groups of coordinate data sets, such as in the case of reaction coordinate runs, each set is specified in the appropriate order. This schema is obviously still under development to add more data types. A full illustration of the schema is provided at <http://www.sdsc.edu/~jpg/nmi/>.

An example XML document for even a small system can be quite lengthy. A full illustration of such a document for a very basic GAMESS input is illustrated at <http://www.sdsc.edu/~jpg/nmi/>. We plan to use the XML documents generated by GAMESS for several purposes. One, which we have briefly mentioned, is to provide a single structured input for our visual interface QMView. Such a document will provide a single access point for all data produced by GAMESS computations. In turn, QMView will be able to generate XML documents as well as "flat" input files for GAMESS computations. We plan to utilize the JAVA based "Castor" [19] project to "unmarshall" the XML document data from XML documents to JAVA objects. Subsequently, one could imagine using such document data in QMView for processing, or perhaps to prepare new input files for generating several GAMESS runs in sequence (e.g. minimization followed by vibrational analysis). The glue that we plan to use to bind data extracted by the XML documents by the Castor JAVA classes to QMView is provided by the JAVA Native Interface (JNI) [20]. Such an interface will allow us to load the JAVA classes directly into QMView (written in C).

For purposes of utilizing the computational grid, the GAMESS XML documents will provide the basis for transferring data from remote resources to a user from one or more jobs scattered across more than a single resource. This format of data will be ideal for depositing results into remote databases, and/or for creating new instances of GAMESS jobs that will be launched across the grid. The data used for subsequent runs may also be obtained by querying databases, in which case the XML document is the best format for the retrieved data.

Our concurrent project involving the construction of a rigorous federated QM database will serve as a focal point for creating computational chemistry jobs. The database will contain the data addressed by our XML schema along with other calculated properties. As with the GAMESS output, we will bind QMView to the database with JNI. One could imagine exploiting such a database to extract common properties within a single molecular structure or across a group of compounds. For example, given the presence of a 3-D grid of electrostatic data, one could query for certain features such as zero flux surfaces. Interactive queries could also be done on a portion of space occupied by a structure, such as a reactive ligand, and surface volumes in that region could be displayed. Across several compounds, trends for certain properties (structural feature, dipole moment, or vibrational frequency) could be searched and correlated.

4 Conclusions

Computational modeling in the era of grid technology is still very challenging and often confined to specialized research groups that are set up with all the key components and tools. Much of the success in these studies have been despite the difficulties involved in integrating all of the technologies, software, and other tools necessary to answer the scientific questions of interest. Additionally, the current technical software available for general utility is typically limited in computational flexibility, graphics and database capabilities, and access, particularly for the remote user. The effort involved in keeping abreast of latest advances in computer technology, the explosive growth in the size and types of chemical databases, and new visualization and analysis tools, all of which offer new computational solutions for increasingly complex problems, requires an inordinate commitment of time on the part of a researcher who may not be well versed in one or more of the resource areas, nor may they feel obliged to take the time away from research to learn.

Our present goals involve the creation of tools at the interface of molecular electronic structure and associated informatics tools, cheminformatics, and grid technologies, to develop a “one-stop” set of web services to offer the research and education communities. The work described here is ongoing and all the component tools are by no means at their final stages. The extensibility of XML makes it relatively easy to add new data types, making it possible to query and display a wide range of data. Coupled with the Grid Workflow environment, which is also currently under development in this group, efforts expended to the collection of data, running of jobs, analysis, visualization, and dissemination of results will be greatly reduced. The resulting set of tools and resources should enable new thought processes in scientific discovery, as well as be quite important for education and training in the classroom.

Acknowledgements. The authors would like to acknowledge support from the NSF through ANI-0223043, and the NIH through NBCR-RR08605 (NIH).

References

- [1] Baldrige, K. K., Greenberg, J. P., Elbert, S. T., Mock, S., Papadopoulos, P., "QMView and GAMESS: Integration into the World Wide Computational Grid," presented at Proceedings of Supercomputing 2002, 2002.
- [2] Fallside, D. C., XML Schema Part 0: Primer, W3C Recommendation,
- [3] Schmidt, M., Baldrige, K. K., Boatz, J. A., Elbert, S., Gordon, M., Jenson, J. H., Koeski, S., Matsunaga, N., Nguyen, K. A., Su, S. J., Windus, T. L., Dupuis, M., Montgomery, J. A.:The General Atomic and Molecular Electronic Structure System, *J. Comp. Chem.*, vol. 14, 1993 pp. 1347–1363.
- [4] Rajasekar, A. K., Wan, M., "SRB and SRBRack- Components of a Virtual Data Grid Architecture," presented at Advanced Simulation Technologies Conference, SD, CA, 2002. .
- [5] Thomas, M., Mock, S., Dahan, M., Mueller, K., Sutton, D., Boisseau, J. R., "The Gridport Toolkit: a System for Building Grid Portals," presented at 10th IEEE International Symp. on High Perf. Comp, 2001.
- [6] Foster, I., Kesselman C.:Globus: A Metacomputing Infrastructure Toolkit, *Intl. J. Supercomputing Applications*, vol. 11, 1997 pp. 115–128
- [7] Perlman, L. W. V., Foster I., Kesselman C., Tuecke S.:A Community Authorization Service for Group Collaboration, *Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, 2002
- [8] Pezvnar, P. A., Tang H., Waterman M.S.:An Eulerian path approach to DNA fragment assembly, *Proc. Natl. Acad. Sci.*, vol. 98, 2001 pp. 9748–9753
- [9] Ewald, T., Understanding XML Web Services The Web Services Idea, <http://msdn.microsoft.com/webservices/understanding/readme/default.aspx>
- [10] Murray-Rust, P., Rzepa, H. S., Wright, M.:Development of chemical markup language (CML) as a system for handling complex chemical content, *New Journal of Chemistry*, 2001 pp. 618–634
- [11] Baldrige, K. K., Greenberg, J. P.:QMView: A Computational 3D Visualization Tool at the Interface Between Molecules and Man, *J. Mol. Graphics*, vol. 13, 1995 pp. 63–666
- [12] Baldrige, K. K., Greenberg, J. P.:QMView as a Supramolecular Visualization Tool, in *Supramolecular Chemistry*, J. Siegel, Ed.: Kluwer Academic Publishers, 1995, pp. 169–177.
- [13] Greenberg, J. P.:MDMovie: a Molecular Dynamics Viewing Tool, *J. Mol. Graph.*, vol. 14, 1996 pp. 287–290
- [14] Jimenez, J., Salazar, G., Baldrige, K. K., Romesberg, F. E.:Flexibility and molecular recognition in the immune system, *Proc. Natl. Acad. Sci.*, vol. 100, 2003 pp. 92–97
- [15] Baker, N. A., Sept, D., Joseph S., Hoist J. M., McCammon, J. A. J.:Electrostatics of nanosystems: application to microtubules and the ribosome, *Proc. Natl. Acad. Sci.*, vol. 98, 2001 pp. 10037–10041
- [16] Shindylov, I., Bourne, P.:Protein structure alignment by incremental combinatorial extension (CE) of the optimal path, *Protein Eng.*, vol. 11, 1998 pp. 739–747
- [17] McCulloch, A., Continuity 6.0
- [18] Laboratory for Applied Pharmacokinetics, http://www.usc.edu/hsc/lab_apk
- [19] Exolab Group, Castor 0.9.4.2
- [20] Liang, S., *The JavaTM Native Interface: Programmer's Guide and Specification*, 1 ed: Addison Wesley Longman, Inc, 1999.