Easy and Effective Computing Environment on the WWW

Alan Shimoide[1], Luping Lin[1*], Tracie-Lynne Hong[1] , Sergio Aragon[2], Ilmi Yoon[1]

[1]Computer Science Department
[2]Department of Chemistry and Biochemistry
San Francisco State University

[1*]Currently at computer Science Department
Stanford University

Abstract

Computational methods have become a primary tool in the investigation of biological, chemical and physical processes. The computation chain needed for these scientific computations often consists of several separate programs by different authors on various platforms and often requires 3D visualizations of intermediate results. Due to the complexity, tools developed by a particular research group are not readily available for use by other groups, nor even by the non-experts within the same research group. To alleviate this situation, and to foment the easy and wide distribution of computational tools worldwide, we developed an Easy and Effective Computing Environment (EECE) on the WWW including interactive 3D visualization that can be used with any web browser. Java based technologies were used to provide a platform neutral, user-friendly solution. Java Server Pages (JSP), Java Servlets, Java Beans, JOGL (Java bindings for OpenGL), and Java Web Start were used to create a solution that simplifies the computing chain for the user allowing the user to focus on their scientific research. EECE hides complexity from the user and provides robust and sophisticated visualization through a web browser.

1. Introduction

Computational methods have become a primary tool in the investigation of biological and physical processes. The immediate requirement in the development of a new computational tool or software is the ability to program a computer in C, Fortran, or another high-level language. However, most scientific researchers needing these tools are not programmers, and the ultimate determinant of the utility of software tools is how easy they are to use in practice. Much of this software development occurs within particular research groups whose work is published in scientific journals, however, the tools developed by a particular research group are not readily available or easily useable by other groups, nor even by non-experts within the same research group. Within the same research group it's not an issue of internal restrictions, but simply of lack of knowledge to use what are often arcane computational environments developed by the programmers in the group. In order to make software more universally useable, much specialized effort is devoted to the development of graphical user interfaces for each software package. Furthermore, some tools are constructed to take advantage of specific hardware features not generally available, further limiting the use of the software by other people. These circumstances result in the unnecessarily restricted use of software, the duplication of effort in the development of the tools themselves, and the expensive and resource intensive construction of unique user interfaces for the developed tools.

A general purpose tool that can address these types of limitations will greatly enhance research productivity and biomedical researchers will not have to wait for the commercialization of a software package in order to have something that is easy enough to use. The primary feature of the envisioned EECE Application is a computational environment that allows the user to run arbitrary codes, programmed in arbitrary languages, located at an arbitrary location in a network, all within a unified user interface that allows for flow control and visualization of the computational results of any module in the computational chain. The web browser has been selected as the primary user interface because it is ubiquitous and can be enhanced for visualization by universally accessible technologies such as Java.

The scope of project presented in this paper is the development of a tool, an *example* EECE application, to carry out the complex series of computations in biomolecule hydrodynamics. This preliminary work demonstrates that it is possible to construct an easy to use interface accessible via a web browser to carry out a complex computational chain.

The Aragon research group has developed, with NIH funding over the past 4 years, a set of computer programs for the precise computation of diffusion tensors and intrinsic viscosity of biomolecules, with emphasis on the application to proteins and nucleic acids[13]. The computation chain for this type of calculation is a very good example of the difficulties for the general user mentioned previously: the modules are designed to run with arguments from the command line in a Linux operating system, and there are no intrinsic visualization tools within the chain. It is difficult for beginning researchers to use these tools, and sharing the software with other research groups is also inhibited by the lack of graphical user interfaces.

To alleviate these situations, and to foment the easy and wide distribution of computational tools worldwide, we are developing a Easy and Effective Computing Environment on the WWW. When the interface is presented via a standard web browser, which is ubiquitous and a part of every researcher's work environment, researchers take advantage of powerful and possibly remotely located software resources by a few mouse clicks. Hiding the operational complexity from the user promotes the uses of computational resources, collaboration and education.

This project is focused on demonstrating the usability of sequences of computing components accessed through WWW with their virtual workspace and visualization. This allows researchers to focus on their scientific work instead of configuring and managing individual programs. The automatic generation of this kind of application is under development and is not discussed in this paper. Section 2 discusses the background of Hydrodynamic computing, section 3 discusses the implementation of EECE, and section 4 concludes with brief discussion about the on-going automation project.

2. Background

Proteins are long chains of amino acids that have a definite conformation in three dimensions after the chain of amino acids has been folded in a specific fashion. Proteins are essential constituents of living cells and the shape of each protein is vital to the function of the protein. Many protein structures have been deduced using X-ray crystallography and their atomic structures have been recorded in the Protein Data Bank using the pdb file format. X-ray crystallography measures the shape of proteins as crystals but proteins do not normally exist as crystals. Proteins normally exist in an aqueous environment and the size and shape of proteins can differ between a protein in an aqueous environment and a protein in its crystallized state [Gia92]. The hydrodynamic transport tensors of a molecule directly relate to the diffusive properties of the molecule and can be used to ask questions regarding the applicability in solution of the structures derived from X-ray crystallography. Accurate and effective hydrodynamics methods are usually computationally intensive and the recent advancement of computing power may play a crucial role in the advancement of hydrodynamics.

Computational science, including hydrodynamics computing, consists of many complex computation components developed by different authors on various platforms. As computing power increases, problems or approaches that were not easily tackled before become solvable, and consequentially require larger and more complex computation modules to solve. Due to the complexity, tools developed by a particular research group are not readily available for use by other groups, nor even by the non-experts within the same research group.

The current computing chains used for hydrodynamics computing within the Aragon group consist of three programs: MSROLL, COALESCE, and BEST. X-ray crystallography results only contain the atomic structure and not the molecule's surface. MSROLL, developed by Connolly, generates a triangle mesh representing the surface. COALESCE, a Fortran program makes the output of MSROLL suitable for boundary element computations by reducing the numbers of triangles in the mesh and eliminating slender triangles. Researchers view the results from COALESCE several times and revise the input parameters for COALESCE until suitable results are obtained. BEST calculates the hydrodynamic transport tensors of rigid molecules with stick boundary conditions from the output of COALESCE.

Running these programs with various parameters/input data is a complex process. Researchers usually have to execute each computing chain several times, compare intermediate results and iterate the whole computing chains until trustworthy results are obtained. By shielding users from operational complexity, researchers can focus more on their scientific research and less on the mechanics of using and configuring resources. We provide a virtual web-based workspace where each researcher can experiment with the computing chains through consistent execution interfaces. Inside their virtual web-based workspace, researchers can save individual settings and store and refer to execution histories. Execution histories record the input data with their parameters to facilitate easy comparisons and validation of computation results.

Scientific Visualization has become an indispensable part for most computational science since visualization transforms abstract information

into interactive imagery, effectively enhancing researcher's understanding of the problem at hand. Visualization plays a vital role in web-based hydrodynamics computing not only for providing insight, but also for validating intermediate results. The implementation of a precise computational method must carefully handle possible sources of error induced in the process of the generating the discretized surface. There are fundamental sources of error such as discretization, shape and area error that one must consider in a precise implementation of the boundary element method [Ara04]. Validation of intermediate data requires visual assessments of 3D geometry or triangulation of 3D structure of molecules, requiring a 3D viewer that effectively enhances 3D perception of the geometry along with various analysis of the data. We developed a interactive 3D viewer to analyze and visualize 3D molecular geometry data generated from the computing chains with various shading and navigation options on the WWW as a part of EECE.

3. Implementation

This section focuses on the three main components of EECE: current computation chains, 3D visualization, and current design implementation. A brief description of the current computation chains used in EECE will be discussed first, followed by a discussion on the 3D visualization tool developed for the WWW and the technology involved with this tool, and finally a discussion of the EECE design as an Internet Application.

3.1 Current Computation Chains

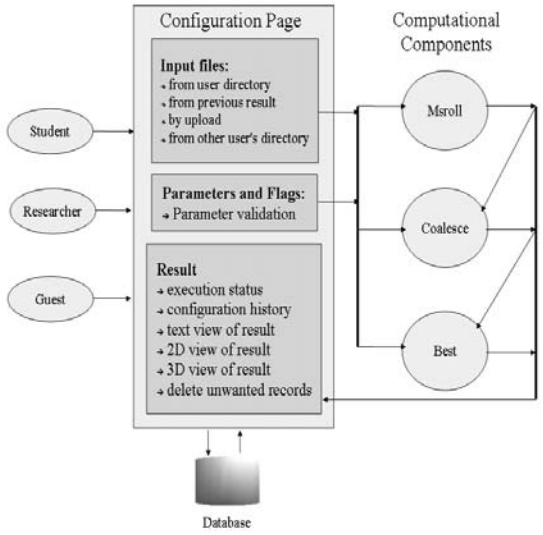

Figure 1: Flowchart layout of EECE.

A triangulation of the surface of the molecule in question is necessary for hydrodynamic calculations using the YA method. PDB files, which are the results from X-ray crystallography, contain only the atomic structure of a protein and not its molecular surface. Connolly has developed a program named MSROLL which calculates the solvent accessible surface by first simulating a ball about the size of the solvent molecule, and then rolling it over the Van Der Waals surface of the atomic structure. MSROLL then generates a triangulation of the molecular surface which has typically been visualized using common mathematical packages. However, this triangulation is not suitable for boundary element calculations since it contains both slender triangles as well as triangles with small areas, which increases the error introduced into the calculations.

Aragon has created a Fortran program, COALESCE, which reduces these problematic triangles as well as reduces the total number of triangles using during surface triangulation. Researchers can view each of the triangulated surfaces generated from COALESCE as they revise their input parameters until suitable results for boundary element calculations are obtained. Once a suitable surface triangulation is generated, Aragon's Fortran program BEST is used to calculate the transport sensors. Using the boundary element method for stick boundary conditions, BEST calculates the protein's hydrodynamic interaction sensors. Figure 1 demonstrates this flow of data through the computation chain. Notice that at each step within the process, iterations can occur to generate a suitable output.

3.2 3D Visualization Tool on the WWW

Having an interactive and intuitive 3D visualization tool is an important component of EECE (Web-based Interactive Computing Environment). A molecule viewer was developed so that researchers can visualize intermediate vertex data from MSROLL and COALESCE over the Web while at the same time providing users with customizable shading and lighting functionalities to aid in both the researcher's perception of the 3D geometry and the discovery of problematic triangles. The molecule viewer has several functions that can help the user correctly understand a molecule's complex 3D geometry. The JOGL (Java OpenGL binding) viewer provides several colored lights and two white lights that can be turned off and on. These lights can be positioned independently of each other around the molecule as desired by the researcher (Figure 2). The JOGL viewer provides a profound new tool to help

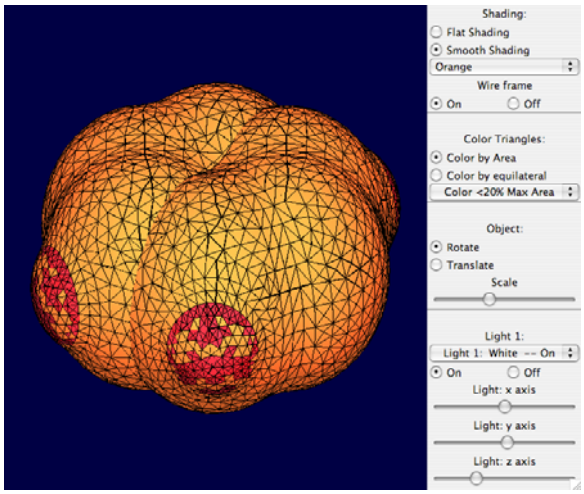

Figure 2: SWING Interface that allows for various visualization options. Image shown has the Smooth Shading and Wire frame features turned on as well as various lights used to illuminate the entire molecular structure.

researchers visualize the source of errors in the triangulation of the molecule directly from the WWW by utilizing Java Web Start (JNLP) and JOGL (Java OpenGL Binding).

3.3 EECE Design

Prior to the development of EECE, researchers looking to generate the structure of a molecule would have to execute a series of programs, oftentimes more than once, through the command line. EECE was designed to simplify this process by providing researchers with an easy to use interface to these same programs that can be accessed from any standard web browser. The layout of EECE is clean, simple, and intuitive which allows even the novice user to easily navigate through the system, configure, and execute their files, as seen in Figure 3. Buttons and links to important features of EECE remain static and its position does not change. This provides users with a static layout regardless of the dynamic content contained within it. In order to support a consistent layout across web browsers, EECE uses the platform-independent Java technologies of JSP, Java Servlets, and Java Beans as well as a MySQL database to store its information.

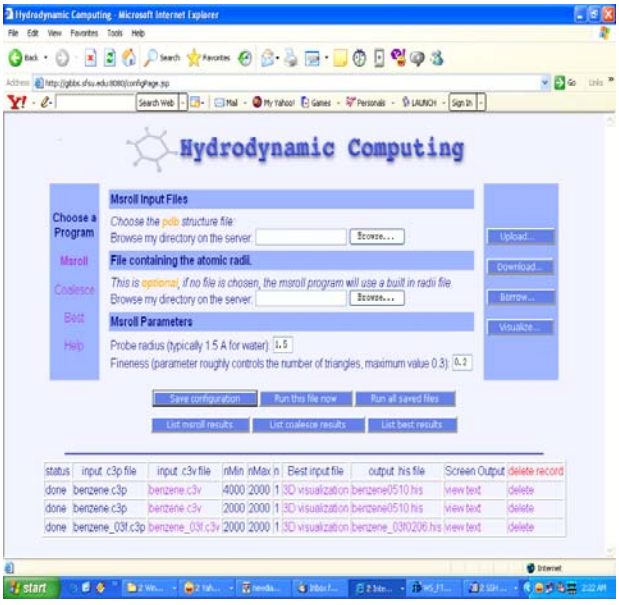

Figure 3: Screenshot of the main WICE interface.

While developing EECE, special attention was paid to the needs of those who would be using the application. Two features that researchers designated as being of high importance were support of virtual work spaces, and a configuration pipeline to store all configurations and results that have been saved or executed. Implementation of these features requires that users provide their user name and password before being given access to any files. This is done through a simple login page where, upon account validation, researchers can access any of their own files which have been previously saved, execute new files, or visualize previous program executions.

Because the area of hydrodynamic computing is a highly competitive field of study, researchers in this area want to have their findings protected from others, while at the same time, have the flexibility to share their data in order to promote research and community growth. In order to provide support for this ideal, EECE provides each of its users with individual virtual work spaces. These workspaces allow the researcher to specify which files, if any, he or she would like to make available to the outside world. For example, suppose a researcher generates the structure for a molecule that has not been generated previously. He/she may not want to publicly share the parameters used to create this structure, but he/she might want to share the basic data used for this generation. EECE would allow this researcher to designate which files to share and which to protect; thus giving him/her full control over his/her files in much the same way as storing the files on his/her own computer and executing the program from the command line would.

Having this work space for user files is only helpful if the user is able to save pertinent data into it. Previous to EECE, researchers would need to execute each program from the command line and even then, they oftentimes had to execute a program multiple times with different parameters before generating the correct result. Researchers had to manually store their

program parameters and results because no system existed to handle this for them. EECE solves this issue by automatically storing both the results and the parameters used to generate the results into the researcher's individual work space through our configuration pipeline.

Configurations defined by the researcher are passed through the pipeline in two simple steps. First, the pipeline checks to see if this configuration is a duplicate of any other configuration which the researcher has already sent through it. If it is, then the results of this particular configuration are sent back to the user without the need to execute the given program. However, if this is a new configuration, the proper program will be called to execute. After execution, both the configuration and its results will be saved within the researcher's work space before being sent back through the pipeline. There are many advantages to using this type of pipeline, the first being that it saves on computation time. Because all configurations and their results are saved by the pipeline, the need to execute a program for each incoming configuration is eliminated. Duplicate configurations by the same user can have their results returned without any computational expense. Secondly, since all configurations and results are saved, users can easily pull up these records and see what they have done in the past. There is no longer a need to manually keep track of all past configurations and their results because it is now a mouse click away. This saves the researcher time in that he/she no longer needs to write everything down; instead he/she can use the extra time to develop better representations of the molecule's structure.

Although many other features have been built into EECE such as file uploading; aside from the protein visualization, individual work spaces and history of configuration and execution are the two most important features of EECE. These two features make EECE a powerful tool for researchers looking to do hydrodynamic computing.

4. Conclusion

The example EECE application was designed to simplify the execution of a computational chain by providing researchers with an easy to use interface to these same programs that can be accessed from any standard web browser, instead of executing programs though a text based command line. The layout of the example EECE is clean, simple, and intuitive which allows even the novice user to easily navigate through the system, configure, and execute their files, as discussed in section 3.

Currently the Aragon group started to let students can guest users access the example EECE and collects users opinion on improving usability. The automation of creating EECE applications such as this micro EECE is under development. Baseline is that the original programmer of computation modules goes through simple interview program to generate Interface Description Language using XML notations, and then place them in computation module repository with the computation modules. A EECE application builder allows users to drag and drop those computation modules in editing environment to create the computation flow. The EECE web interface similar to figure 3 will be automatically generated and the application will be launched as a web application. The detail deserves a dedicated paper when completed.

References

[Ant89] Antosiewicz, J.; Porschke, D. J Phys Chem 1989, 93, 5301.

[Ara04] Sergio Aragon, "A precise boundary element method for macromolecular transport properties", J. Computational Chemistry, 25, 1191-1205 (2004).

[Ber76] Berne, B.; Pecora, R. Dynamic Light Scattering: With Applications to Chemistry, Biology and Physics. Wiley-Interscience: New York, 1976.

[Blo67] Bloomfield, V. A.; Dalton, W. O.; van Holde, K. E. Biopolymers 1967, 5, 135; Ibid. Biopolymers 1967, 5, 149.

[Ede83] Eden, D.; Elias, J. G. In Measurement of Suspended Particles by Quasi-Elastic Light Scattering; Dahneke, B., Ed.; Wiley-Interscience: New York, 1983.

[Gar77] Garcia de la Torre, J.; Bloomfield, V. A. Biopolymers 1977, 16, 1779.

[Gar81] Garcia Bernal, J. M.; Garcia de la Torre, J. Biopolymers 1981, 20, 129.

[Gia92] Giacovazzo, C., Ed. Fundamentals of Crystallography; Oxford University Press: New York, 1992.

[Ric80] Richards, E. G. An Introduction to Physical Properties of Large Molecules in Solution; Cambridge University Press: New York, 1980.

[Str68] Stryer, L. Science 1968, 192, 526.

[Swa97] Swaminathan, R.; Hoang, C. P.; Verman, A. S. Biophys J 1997, 72, 1900.

[Tel79] Teller, D. C.; Swanson, E.; de Haen, C. Methods Enzymol 1979, 61, 103.

[Wil79] Wilson, R. W.; Bloomfield, V. A. Biopolymers 1979, 18, 1205.

[Yoo04] *Yoon, I., Williams, R., Levine, E., Yoon, S., Dunne, J., Martinez, N.D.* Webs on the Web: 3D Visualization of Ecological Networks on the WWW for Collaborative Research and Education. SPIE Electronic Imaging conference, 2004.

[You75] Youngren, G. K.; Aerivos, A. J Fluid Mech 1975, 69, 377.