East Main Educational Consulting, LLC
Wilmington, NC

Fostering Undergraduate Research Partnerships through a Graphical User Environment for the North Carolina Computing Grid

Final Report

Prepared and submitted by:

Barbara P. Heath, Ph.D.
East Main Educational Consulting, LLC

Ronald J. Vetter, Ph.D.
University of North Carolina-Wilmington

May 2007

1. Executive Summary

The UNCW Grid Computing Project made significant progress toward the stated goals and outcomes. The longitudinal data collection showed a trend toward responses of significant and very significant progress as it relates to the goals and outcomes. Progress was consistent, and no one goal or outcome was delayed.

Significant progress was made in developing scientific workflows for representing and managing complex distributed scientific computations. These computations included many steps, where each step integrated applications and data sources developed by different groups. The applications and data were distributed in the project's grid execution environment. The GridNexus software and the associated JXPL programming language were able to capture the individual data transformations and analysis steps typically used by scientific workflows as well as the mechanisms to carry them out in a distributed environment. The software has been approved to be released as open source software at the close of this project (June 2007). It is anticipated that additional software developers and user groups will embrace this workflow system when that happens. At the time of this report, 26 workflows and 7 lesson plans were available for download.

Individual team members reported that much progress was made toward the inclusion of students as key partners to the software development for specific university personnel. The evaluation team observed the faculty-student teams working together and presenting their projects to others. The teams communicated effectively, while learning the terminology of the complimentary content area. Also, the evaluation team observed the Computational Chemistry product successfully tested with students in a course setting. Student programmers reported that the interaction with computer science faculty and scientists influenced their decision to attend graduate school.

A Grid Computing course, the first of its type in the nation, was offered as part of the project training efforts. The course was offered simultaneously at numerous campuses. Fifty-four students enrolled in two course offerings during the project funding cycle. Additional training materials (tools and modules) were successfully created and have been posted for general use. At the time of this report, hundreds of lecture slides, 7 programming assignments, and more than 11 tutorials are available for download. A listing of all related publications and presentations is also available (37 at the time of this report).

Significant Outcomes

- Project software available as open-source
- Additional funding secured through the technology division at UNCW to achieve open-source status and engage the user community
- JXPL and GridNexus are being used by private company for a SBIR phase II project
- Student participation extended well beyond being involved in research
- High percentage of student programmers chose to pursue graduate studies and reported that their interaction with faculty and scientists influenced their post-graduate decision
- Students participated in near-peer roles as exampled by a) the expectation to present progress reports at all-hands project meetings and b) conversations between students and faculty consistently at a collegial level, rather than professor as knowledge-holder

- Grid partners have continued their grid activities and building upon the momentum generated thus far though the NC Grid Computing Initiative especially at NCSU, CFCC, and NCCU
- Numerous opportunities, generated by Grid project activities, limit participation among partners
- Several grant proposals have been written to secure additional funding for future activities directly resulting from the project
- Collaboration between UNC institutions was strong
- New partners have been added both within original partner institutions and non-partner institutions

2. Project Context *(excerpts from proposal)*

Grid computing is one of the most intriguing areas in the computer science field and is likely to have a significant impact on the scientific and business arenas in the near future. A Grid is a network infrastructure that enables collaborative use of distributed resources, using general-purpose protocols and interfaces.

At the time of the proposal, interacting with Grid services was described as a complex process. Most of the existing applications that access grid services required the user to type cumbersome commands, often using a command-line interface. Creating new clients and services requires programming in a language such as Java or C and using a host of libraries for interacting with the Open Grid Services Infrastructure, Grid Security Infrastructure, Web Services Description Language and other standards. These tools and techniques are useful to a select group of specialists; however the only way to make Grid resources accessible to a wide range of users is to provide a relatively simple graphical user interface (GUI). This project proposed to develop a Grid interface that was easy-to-use and used by a wide-range of applications and users.

Prior to project funding, UNCW had developed an innovative GUI for grid applications. In particular, they introduced a new scripting language designed for web-based services, a GUI for creating scripts, and demonstrated the use of these tools with grid services. In the prototype GUI, the user selected modules from a library and dragged them to a work area. A module represented a Grid service client or some action based on the result of a service call. For example, one module might represent a service that accessed a database, and other modules might filter the data or perform some statistical analysis.

The aim of the funded project was to develop the Grid GUI and provide a unified view of Grid resources, services and their interactions, much like a browser provides a way to visualize and use the World Wide Web. This GUI can help to facilitate the adoption of grid technologies in a way very similar to the way the web browser helped to foster the development of the web. It took the graphical nature of the browser for users to understand and visualize the power of the web and how they could make use of it.

The project was a collaborative effort among a multi-disciplinary, multi-investigator UNCW core research team and several discipline-focuses researchers at a variety of NC institutions. Disciplined focused teams included:

- Software Development Efforts (UNCW)
- Computational Chemistry (UNCW, ECU)
- Bioinformatics (UNCW, NCSU, UNC-CH)
- Combinatorics (UNCW)
- Business Computing (UNCW, NCCU)
- Education and Training (UNCW, CFCC, WCU, UNC-CH)

3. Project Goals *(excerpts from proposal)*

The Grid Computing Project was grounded by two basic principles: (1) the development of an infrastructure of real applications and working solutions, and (2) the creation and use of grid software technologies that can adapt to diverse and ever-changing environments. The project overview document describes the vision of the project as follows:

> *"By partnering faculty researchers with undergraduate student programmers, we intend to develop a wide range of system tools and user library modules for a variety of tasks commonly needed for computational chemistry, bioinformatics, human-computer interfaces, workflow programming, combinatorics, and e-business research. We intend to make the UNCW Grid GUI available as open source software and to share our partnership modules freely with others across the state through workshops and presentations, hands-on training, and through the development of online educational resources. We expect this project to have statewide impact and national prominence."*

To achieve this vision, the project team set five goals and four broader outcomes to be achieved during the two year project.

Goals:

1. To move from a prototype Grid GUI and JXPL interpreter to production versions
2. To create additional system library tools and user library modules for the UNCW Grid GUI and to share them with others
3. To leverage the talents of undergraduate computer science and information systems students by partnering them with faculty scientists who have custom software development needs
4. To develop collaborations with industry, research organizations, government and state agencies, and other education institutions
5. To help support the development of the North Carolina Grid Initiative by developing a series of workshops, presentations, and online resources to train students, faculty, and business personnel to use grid technologies

Broader Outcomes:

1. Encourage entrepreneurial activity among faculty, staff, and students to create new innovations and help to spark economic development across NC
2. Involve undergraduate students in research
3. Help support other grid partners to continue their grid activities and to build upon the momentum generated thus far though the NC Grid Computing Initiative.
4. Foster collaboration between UNC institutions, industry, and other government and state agencies

4. Progress Details

The progress of the Grid Computing project will be summarized by the goal statements, albeit not in the order in which they were originally presented. In order to describe the complexities of moving from a prototype GUI and JXPL interpreter to production versions, two separate processes will be described. First the technological development and second the collaboration between the undergraduate students and science faculty.

4.1 Technological Development Process

4.1.1. To move from a prototype Grid GUI and JXPL interpreter to production versions

The project's software development team adopted a waterfall development model. The waterfall model is a sequential software development model (a process for the creation of software) in which development is seen as flowing steadily downward (like a waterfall) through the phases of requirements analysis, design, implementation, testing (validation), integration, and maintenance.

The software developers also focused on supporting strong team collaboration. To coordinate the efforts of each team, the project employed the open source Subversion (SVN) system. Subversion is a revision control system which allows computer software to be developed in an incremental and controlled fashion by a distributed group of programmers. The log files of this system allow us to monitor both individual and team performance. During the software development lifecycle, the team adopted a proven model of the software evolution process – the model of incremental change (IC). In this model, IC is initiated with a change request that originates from a project stakeholder who requests new functionality to be incorporated into the existing system. A change request may take the form of a bug report, which requires a certain functionality to be corrected, or it may stipulate a completely new functionality that is demanded by user and/or technological need. In this project, change requests also came from a "wish list" of functionality that was prepared by the faculty team that was leading the software development effort.

Unit and functional testing was performed to ensure that all classes affected by a change request are correct. Additional testing was performed to ensure that the old functionality remained intact after a change had taken place and to check the correctness if a new functionality was added. Finally, software documentation was produced by using the Javadoc tool for generating API documentation in HTML format from doc comments in system's source code.

The GridNexus software, and JXPL specification, has been approved to be released as open source software. By the end of the project funding (June 30, 2007), the SVN code repository for GridNexus will be made available to the public on SourceForge.net – the world's largest development and download repository for open source software applications. Thus, the project will have successfully migrated from a prototype version to a production version.

4.2 Collaboration

The collaboration efforts of this project were at three levels, between the:

- undergraduate students and science faculty
- discipline-focused teams
- beyond funded partners

Overlaps in the differing levels of collaboration were evident, which increased the rate of progress and strengthened the project results. The collaboration between and among these groups assisted in the attainment of the remaining four goals. Details of the highlights for each goal are summarized in the following sections.

4.2.1 To leverage the talents of undergraduate computer science and information systems students by partnering them with faculty scientists who have custom software development needs

Multiple undergraduate-faculty groups were arranged to accommodate each of the discipline-focused teams. The evaluation focused on two development groups, Bioinformatics and Computational Chemistry, to gain an understanding of collaboration between the science faculty and computer science undergraduate students. Mid-way through the project, a survey was deployed to collect information from the students involved in each of the discipline-focused teams.

The Bioinformatics group meetings were observed during fall 2004. The meetings incorporated brainstorming. The conversation between the faculty and students prompted ideas and possibilities. A leveling of the playing field was obvious. The project was investigating a new technology, so everyone was learning as the project moved forward. Science faculty members were learning the computer science aspects and students were learning both computer science and the science discipline. If this did not occur, communication between the groups would breakdown. A student was noted as saying "we need to break the language barrier..." referring to specific words having different meaning in different disciplines. The computer science faculty team members did not attend the group meetings; however, the students made notes of what needed to be reported back to these members- whose role was focused on the software development. By the end of the semester, the groups' progress slowed while awaiting the resolution of multiple software and hardware issues. This prompted a shift in focus from the Bioinformatics group to the Computational Chemistry group.

The Computational Chemistry team had made great strides in the development of the GUI interface aimed to run *DMol3* and *Gaussian* over the Grid, which will make it easier to incorporate the computational packages into the classroom and for computational chemistry researchers to access the software packages. The GUI was ready for initial implementation in the classroom in April 2005. A lab that students complete each year was adjusted to provide the opportunity to test the interface with new users. The instructor planned the lab around the two shortcomings of the current system: a) submission/calculation time-out is about three minutes (calculations requiring more than three minutes will not be completed), and b) users can only submit one job at a time. The current limitations were rectified after current versioning updates.

The observation session included three students (2 males, 1 female) and began with the standard pre-lab briefing, which included a short demonstration of how to use the interface. Students then moved to the lab and worked independently to complete their calculation. The instructor was on hand to answer questions and provide reminders. Most students did not take notes during the demonstration, but were able to re-create the process with little input from the instructor. After students collected their results, they participated in a short focus group regarding the usability of the interface.

Students easily recited the steps necessary to complete the activity using the new interface, correcting the order of two of the steps during the discussion. They recited the steps without looking at their computer or notes. Students reported that the steps were easy and rated them a two on a scale from 1-10 (easy→difficult). They also shared that the experience was not difficult, the screen text was simple to understand, and the interface was easy to use.

Observing this particular group of students was helpful since they were not only the first group to use the interface, but the previous week they had to complete the lab using the "old" method. This provided an opportunity to compare the old and new methods. When students were asked what they liked about the new interface they replied "it was very simple" and you "don't have to type 100 different commands." Students were surprised at the simplicity of the new method. The previous week students spent approximately one hour completing a similar activity, while with the new method, students were able to finish in five minutes. Most of the additional time was attributed to typing information into different formats, prior to submission for calculation. The new GUI allows the users to by-pass the tedious tasks and focus on the content.

To better understand the impact of the project on the students, a survey was deployed asking a short list of questions related to their experience. Students strongly agreed that they were given the opportunity to create and test new innovations, applied principles learned in their computer science courses, were influenced to do independent research to meet the demands of the project, and faculty guidance was available to assist with the demands of the project. Students did not feel that they had much of an impact in the research direction, but that the interactions with the faculty and scientists influenced their decision to pursue a graduate computer systems degree. The attitude of the students is best summarized in the following student statement:

> *"This project has been an unmatched learning experience for myself and other students. I believe that there should always be ongoing projects involving student researchers to supplement the students' education. I am very appreciative to all who have been a part of this project especially those are most responsible for its success."*

4.2.2 To create additional system library tools and user library modules for the UNCW Grid GUI and to share them with others

The teaming of computer science faculty and students with discipline-focused faculty resulted in the production of several digital resources that are available via the project website (www.gridnexus.org). Items available for use include simple discipline-specific workflows and lesson plans (Table 1). Items posted at the project website were developed and implemented over the course of the two year project. Lesson plans are relevant to Bioinformatics and Computational Chemistry; additional items will be added to the website as they are completed.

Table 1. Summary of simple workflows and lesson plans

Title	Primary Subject	Objective	Category
Arithmetic	General	To configure a generic primitive to modulo	Work Flow
Ceil	General	To divide two numbers and round the value up to the nearest whole number	Work Flow
Compare	General	To compare and see if .3333333 is less than or equal to 1/3. Returns True if it is…. and a blank list if not.	Work Flow
Conditional	General	To perform addition over 2 numbers… if greater than or equal to 10 return value else return the string "Less Than"	Work Flow
Cond	General	To use a Conditional to return YES IF 10 was not less than 10.1 OR 5 was equal to 5/1 otherwise return NO.	Work Flow
Cons	General	To put the two lists together (using the LISP the inspired actor CONS)	Work Flow
Defun	General	To define a function that will double the number passed in. Then invoke it.	Work Flow
Divide	General	To use Arithmetic object to divide two numbers.	Work Flow
Do	General	To run a loop that for 10 iterations adding to the value sum on each pass.	Work Flow
Factorial	General	To create a user defined function for calculating a factorial and invoke.	Work Flow
FileIO	General	To write some text to a file, then read the file and display the content.	Work Flow
First	General	To use the LISP inspired function FIRST to extract the first item from a list.	Work Flow
FullMathService	General	To create, invoke, and destroy a service instance.	Work Flow
FullStickyNote	General	To create a sticky note instance, write a note, show the note and then destroy the instance.	Work Flow
html	General	If the first value is less than 10.1 then return yes, otherwise generate a web page saying hello world.	Work Flow
isEven	General	To check to see if the given number is even.	Work Flow
isPrime	General	To determine if the provided number is prime or not.	Work Flow
JxplService	General	To create, invoke, and destroy a JXPL Service instance on our demo server.	Work Flow
Logic	General	To perform a negated comparison.	Work Flow
MathService	General	To load EPR from file (run MathService2Create first if nonexistent). To add onto the stored service instance and return the current value.	Work Flow
Modulo	General	To perform a modulo operation using the modulo actor.	Work Flow
Prog	General	Use a Prog to set a variable then use the variable for multiplication.	Work Flow
Rest	General	To use the LISP inspired function REST to return a list without the first element.	Work Flow
Setq	General	To store a number in a variable named "X" then add it unto itself.	Work Flow
Sum	General	Basically, to add 2 numbers.	Work Flow
Symbol	General	To add 2 numbers together, save the result and reuse it for multiplication.	Work Flow
Using Mfinder with GridNexus	Biology	To show how items, such as genes can be examined for significantly enriched connections between these items using the Mfinder algorithm in GridNexus	Lesson plan and workflow
Submitting a Gaussian Calculation on the Grid using	Chemistry	To learn how to submit a quantum mechanical calculation using Gaussian software on a computer on the grid with GridNexus	Lesson plan and workflow

GridNexus			
Converting a .pdb file to a Gaussian .dat input file using GridNexus	Chemistry	To learn how to convert a .pdb or .ent file created by another program into a Gaussian input file (.dat) for use within a GridNexus workflow	Lesson plan and workflow
Using Weka with GridNexus	Biology	To use data-mining methods packaged in Weka to interactively search for patterns and associates within the data set	Lesson plan and workflow
Using TargetP with GridNexus	Biology	To use the TargetP algorithm to search for the sequence and structure characteristics in protein sequences that indicate which cellular compartment is the 'home' of a protein	Lesson plan and workflow
MolStart on the Grid	Chemistry	To learn how to use MolStart within GridNexus to construct molecules, edit molecules or load molecular structures and then use GridNexus to run a quantum mechanical calculation on the Grid	Lesson plan and workflow
Using Phylographer with GridNexus	Biology	To show how DNA sequences can be compared to each other using the BLAST algorithm; and then how the resulting matrix can be interactively visualized using the Phylographer program within GridNexus	Lesson plan and workflow

4.2.3 To help support the development of the North Carolina Grid Initiative by developing a series of workshops, presentations, and online resources to train students, faculty, and business personnel to use grid technologies

The first item of business related to education and training was with the students. A Grid Computing course was offered for the first time in the fall of 2004. This course was one of the first of its kind in the nation and was offered to multiple universities and community colleges in North Carolina. Forty students enrolled in the course, including many students who were involved in the project.

At the conclusion of the course, a survey was distributed via the Web. Twenty-one students responded to the survey. For the most part, students were satisfied with the course and would recommend the course to others. Students agreed that they took responsibility for their own learning and that their understanding of key concepts and principles was enriched. Nearly all students included free responses to the final two questions: a) The most valuable aspects of this course, and b) Specific recommendations to improve this course. Commentary was forwarded to the course instructors to inform course revisions. Many comments included the difficulty in working with a new technology. Comments were best summed with the quote, "…the frustration that I experienced tackling the assignments closely parallels the frustration using 'grid technology' in the 'real world.'"

The Grid Computing course was offered for the second time in fall 2005. Thirty-three students representing nine campuses enrolled in the course. At the conclusion of the course, a survey was distributed via the Web. Sixteen students responded to the survey. Students ranked the course as "good" and would recommend the course to others. Students agreed that they took responsibility for their own learning and that their understanding of key concepts and principles was enriched. They also reported that the course materials were of high quality and assisted with their learning. Student responses were equally distributed across the response scale (strongly agree → strongly disagree) on two items: feedback provided by the instructors was helpful and interaction with classmates facilitated learning. Nearly all students included free responses to the final two questions: a) The most valuable aspects of this course, and b) Specific recommendations to improve this course. Students reported that the most valuable aspects of the course include the overview of grid computing, learning the jargon, hands-on experiences, and guest speakers. Students also provided their impressions on how the course could be improved. Suggestions include requiring a group project for completion during the semester, more depth to the topics, and increasing communication opportunities between students to facilitate interactions outside of class. Commentary was forwarded to the course instructors to inform course revisions.

4.3 Online web links and resources

4.3.1 General tutorials and links to programming in C

- JXPL Language Tutorial
- GridNexus Software Tutorial
- Computational Chemistry Tutorials (Lee Bartolotti)
- Basic Configuration Tutorials (Jeff Brown)
- Grid Portal Developers Workshop (Lawrence Berkeley Lab)
- The Globus Toolkit 4 Programmer's Tutorial (Borja Sotomayor)
- GT4 Security Examples (Jeff Brown)
- North Carolina Grid Workshop (June 22-24, 2005)
- Deploy a C application as a grid service
- Programmming in C/C++ with the Java Native Interface
- OGSA-C Programmer’s Guide

4.3.2 Presentations and Publications

Brown, J., Freeze, M., Harris, E., & Vetter, R. (2007, May 7). Parallel Processing with GT4, Poster Presentation at the International Conference on the Virtual Computing Initiative (ICVCI 2007), Research Triangle Park, North Carolina.

Harris, E. & Vetter, R. (2007, April 13). GridNexus: A Graphical User Interface for Grid Computing Applications, Grid Computing Symposium and High Performance Computing Workshop, NC A&T State University, Greensboro, North Carolina.

Wilkinson, B.(2006, April 25). VisualGrid: An Infrastructure for Visualization and Environmental Research Fostering Collaboration and Re-source Sharing. Collective effort of UNC-C, UNC-A, and the EPA. Panel 2, SIMVac Symposium, Asheville, North Carolina.

Wilkinson, B. (2006, February 22). SURAGrid and Grid Computing Activities at UNC-Charlotte. SURAGrid Project Meeting, Washington, DC.

Wilkinson, B. (2005, December 9). Experiences using NCREN to Teach Grid Computing. NCREN Community Day event at MCNC, Research Triangle Park, North Carolina.

Martin, N., Vetter, R., Ferner, C., Brown, J., Martin, A., Martin, P., et. al. (2005, November 1-2). GridNexus, a Graphical User Interface for Grid Computing. SE-SW Regional Meeting of the American Chemical Society, Memphis, Tennessee.

Livingston, J., Yancy, S., Hill, S., McClelland, L., Abou-Chacra, R., Jackson, N., et. al. (2005). Psychosocial Stress, Institutionalized Racism and Susceptibility to Cardiovascular Cardiovascular Disease among African Americans. Southeastern ECO Conference Social and Cultural Dimensions of Health 2005.

Hudson, T., Stapleton, A., & Curley, A. (2005, August 8-11). Minimal Marker Sets to Discriminate Among Seedlines. IEEE Computational Systems Bioinformatics 2005, Stanford, California.

Stapleton, A. (2005, June 16-17). A Scientist's Perspective on Distributed Access to Grid Computing Resources. Identity and Access Management Workshop, Chapel Hill, North Carolina.

Wilkinson, B., Holliday, M., & Ferner, C. (2005, May 9-12). Experiences in Teaching a Geographically Distributed Undergraduate Grid Computing Course. The Second International Workshop on Collaborative and Learning Applications of Grid Technology and Grid Education (Held in conjunction with CCGrid2005), Cardiff, UK.

Martin, N. (2005, April 18). Integrating Computational Chemistry into the Undergraduate Chemistry Curriculum at UNCW and Grid Computing at UNCW. Joint Seminar with Departments of Computer Science and Chemistry, Duke University, Durham, North Carolina.

Shipman, W., Rawls, A., Curley, A., Vetter, R., Ferner, C., & Brown, J. (2005, April 12). GridNexus: A Grid Services Scientific Workflow System. Presentation to the North Carolina General Assembly, Raleigh, North Carolina.

Hunt, C., Ferner, C., & Brown, J. (2005, April 8-10). JXPL: An XML-based Scripting Language for Workflow Execution in a Grid Environment. IEEE Southeastern Conference, Fort Lauderdale, FL.

Wilkinson, B., & Holliday, M. (2005, March 30). State-Wide Collaborative Grid Computing Course. 2005 Teaching and Learning with Technology Conference, Raleigh, North Carolina.

Martin, N., Phillip, J., & Martin, A. (2005, March 29). Development of a User-Friendly GUI to Facilitate Computational Chemistry over a Grid Interact Technology Showcase at UNCW, Wilmington, North Carolina.

Holliday, M., Wilkinson, B., House, J., Daoud, S., & Ferner, C. (2005, February 23-27). Geographically-distributed, Assignment-structured Undergraduate Grid Computing Course 36th ACM Technical Symposium on Computer Science Education (SIGCSE2005), St. Louis, Missouri.

Wilkinson, B. (2005, February 16-18). Introducing Grid Computing into the Undergraduate Curricula. Poster session: NSF Engineering and Computing Education Grantee Meeting, Washington, DC.

Wilkinson, B. (2005, February 11). Grid Computing across North Carolina. UNC-Charlotte Department of Computer Science graduate seminar series, Charlotte, North Carolina.

Brown, J., & Ferner, C. (2005, February 7-11). GridNexus and JXPL: A Grid Services Workflow System. GlobusWORLD, Boston, MA.

Vetter, R., Brown, J., & Ferner, C. (2004, December 10). GridNexus and JXPL: A Grid Services Workflow System. Presentation at MCNC, Research Triangle Park, North Carolina.

McClellan, M. (2004, October 13-15). Emerging Standards for Interoperable Biological Systems. Technology for Life: North Carolina Symposium on Biotechnology and Bioinformatics, Raleigh, North Carolina.

Martin, N. (2004, September 24). Grid Computing: Harnessing Underutilized Resources. UNCW Department of Chemistry and Biochemistry Seminar, Wilmington, North Carolina.

Wilkinson, B. (2004, September 13). Introduction to Grid Computing and Applications in Computational Sciences. Presented at Winston-Salem State University, Winston-Salem, North Carolina.

Wood, M., Ferner, C., & Brown, J. (2004, March 26-28). Toward A Graphical User Interface for Grid Services. Proceedings of the IEEE Southeastern Conference, Greensboro, North Carolina, 316-324.

Brown, J., Ferner, C., Vetter, R., & Wood, M. (2003, October 28-29). UNC-OP and MCNC Grid Computing Forum. Friday Center, Chapel Hill, North Carolina.

Wilkinson, B. & Ferner, C. (2006, July). Teaching Grid Computing Across North Carolina Part II. *IEEE Distributed Systems Online*, *7(7)*.

Wilkinson, B., & Ferner, C. (2006, June). Teaching Grid Computing Across North Carolina Part I. *IEEE Distributed Systems Online*, *7(6)*.

Wilkinson, B., & Ferner, C. (2006). Chapter 156: Grid Computing Implementation and Applications. The Handbook of Computer Networks, H. Bidgolo, Editor-in Chief, John Wiley & Sons, 2006.

Holliday, M., Wilkinson, B., & Ruff, J. (2006, March 10-12). Using an End-to-End Demonstration in an Undergraduate Grid Computing Course. ACMSE 2006: 44th ACM Southeast Conference, Melbourne, FL.

Brown, J., Ferner, C., Hudson, T., Stapleton, A., Vetter, R., Carland, T. et. al. (2005, June 20). GridNexus: A Grid Services Scientific Workflow System. International Journal of Computer Information Science (IJCIS), *6(2)*, 72-82.

Wilkinson, B., Holliday, M., & Ferner, C. (2005, May 9-12). Experiences in Teaching a Geographically Distributed Undergraduate Grid Computing Course. Proceedings of the Second International Workshop on Collaborative and Learning Applications of Grid Technology and Grid Education (Held in conjunction with CCGrid2005), Cardiff, UK.

Hunt, C., Ferner, C., & Brown, J. (2005, April 8-10). JXPL: An XML-based Scripting Language for Workflow Execution in a Grid Environment. Proceedings of the IEEE Southeastern Conference, Fort Lauderdale, FL.

Baker, M., Apon, A., Ferner, C., & Brown, J. (2005, April). Emerging Grid Standards IEEE Computer, *38(4)*, 43-50.

Holliday, M., Wilkinson, B., House, J., Daoud, S., & Ferner, C. (2005, February 23-27). Geographically-distributed, Assignment-structured Undergraduate Grid Computing Course Proceedings of the 36th ACM Technical Symposium on Computer Science Education (SIGCSE2005), St. Louis, Missouri.

Wrede, C., Blum, J., Brown, J., & Stapleton, A. (2004). Comparison of Transmembrane Helix Frequencies using Whole-Genome and Maize Expressed Sequence Curve Fits. *Maydica*, *49(2)*, 67-76.

Hudson, T., Stapleton, A., & Brown, J. (2004, July). Codifying Bioinformatics Processes without Programming. *BioSilico*, *2(4)*, 164-169.

Wood, M., Ferner, C., & Brown, J. (2004, March 26-28). Toward A Graphical User Interface for Grid Services. Proceedings of the IEEE Southeastern Conference, Greensboro, North Carolina, 316-324.

4.2.4 To develop collaborations with industry, research organizations, government and state agencies, and other education institutions

Collaborations with other institutions developed over the course of the two year effort. Most were with other educational institutions; however, relationships with MCNC, a research organization, and IBM and Lexxle, Inc (a software company focused on biologically-inspired and agent-based computing) have also grown. Relationships with project partners North Carolina State University, North Carolina Central University, East Carolina University, University of North Carolina – Chapel Hill, and Cape Fear Community College were expected to grow and did as part of the project activities. These relationships led to additional partners within university partners (plant science at North Carolina State University, Bioinformatics at North Carolina Central University, and the Center for Business and Economic Services at UNC Wilmington). The successes of the project also caught the attention of potential future partners such as the Virginia Bioinformatics Institute and University of Arizona with whom some project partners are pursuing additional funding options. Discussions with government agencies had occurred before the conclusion of the grant activities, but no progress toward definitive partnerships had been reached.

5. Lessons Learned

- The project team was reminded throughout the project that Grid computing is interdisciplinary. Without the discipline specific faculty, there are no real world problems for the computer scientists to solve. Collaboration among the groups is necessary to make Grid computing a reality.
- The interaction of content, education and training, and software development is essential to the development of Grid computing technology. In other words, the development of this technology should be managed as a continuum, not as individual subsets of the greater whole.
- Communication is necessary between the discipline faculty and the computer scientists. Each are experts in their field, but may not use the same language to express the same idea. It is essential for the teams to develop relationships that provide the opportunity to learn the nuances in terminology.

- Working with an evolving technology that is still not widely used is difficult. The tasks require considerable experience and expertise. However, because the technology is so new, it levels the field, allowing more equal collaboration between faculty and students.
- Grid technologies have matured considerably over the past two years. Migration from GT 3.2 to GT 4 was difficult as many specification changed. GT 4 is much easier to work with, but much time was exhausted during year one on GT 3.2.
- Student programmers matched with faculty researchers was of great benefit to this project and the model will be applied to future projects. This interaction benefits both the research project and individual students.
- Undergraduate students have a difficult time learning complex technology in the limited time and they graduate too soon.
- Students require more management, which impacts faculty schedules. However, the benefits to the students and the project vastly outweigh the increased expectations upon faculty.