Multi-Agent Spatial Simulation
A brief discussion with Dr. Munehiro Fukuda about the MASS Library and its real world applications

So, what is MASS?
MASS is a parallel computing library that automatically parallelizes applications so that they are executed over a cluster of multiple computing nodes or a cluster of GPUs. The library our research group is working on is called Multi-Agent Spatial Simulation Library. The basic idea is, we would like to mimic the very complicated phenomena of agents and multiple multi dimensional arrays. For instance, if you would like to write a computational fluid dynamics simulation we would like to compose a 3-D array that actually mimics the air flow. The array should be divided into small chunks and allocated automatically to the different computing nodes so we can perform parallel computing.

If you would like a more complicated example, an influenza epidemic simulation application named FLUTE creates 300 million agents. In this instance each agent represents a man, woman, or child in the United States. Each of the agents interacts with others on a synthesized 2-D array that can be mapped to the US, then it simulates what would happen if a small number contracted the flu and how the flu would spread. This application runs quite slowly because there are 300 million objects, so the application designer has parallelized the application using a message-passing interface. This is a very popular parallelizing library, but is based on message passing between the processors. So, the application had to be drastically rewritten in accordance with MPI Library. Our thought is: what if we allow the application designer to automatically instantiate 300 million objects and those objects would be automatically distributed over a cluster of computing nodes and spaces which were also divided automatically? This way the designers don’t have to worry about underlying parallel computing and execution would be accelerated for faster computation. We were quite interested in how to design applications and how to partition those applications and parallelize them over different computing nodes because this is often quite technical.

The application also includes professor Mike Stiber’s neural network, which is also designed as a multi dimensional array and could automatically divide into small chunks.

Other possibilities include systems biology, which Prof. Wooyoung Kim is working on. We would like to create a protein network as a multi dimensional array. What she would like to find out is a very useful network motif from a protein network. So why don’t we inject as many agents as possible over the array that represents the protein network? The agent will check which

FOLLOW & BEFRIEND US
Stay in touch and up-to-date with what’s going on within the CSS Program. Visit our website (www.uwb.edu/css) for links to our Twitter and Facebook pages.
Dear Friends of Computing & Software Systems,

I have news for you: the CSS Program is no more. Instead, we are now the Computing and Software Systems Division of the new UW Bothell School of Science, Technology, Engineering, and Mathematics. We are joining with the other STEM Divisions — Engineering and Mathematics, Biological Sciences, and Physical Sciences — to create one of the few university campuses where all of these disciplines are housed under one (virtual) roof. What does this mean? Not only will all of the current CSS educational opportunities continue (our BS and MS in Computer Science and Software Engineering, our BA in Applied Computing, and our Graduate Certificate in Software Design and Development), but a whole host of new initiatives will be coming down the pike. This includes our new MS in Cyber Security Engineering (now enrolling for fall 2013), two joint degrees (BS in Computing Engineering, joint with Engineering and Math, and BA in Interactive Media Design, joint with the UWB School of Interdisciplinary Arts and Sciences), a tri-campus undergraduate concentration in Information Assurance and Cybersecurity, and may more degrees, minors, and concentrations over the next few years.

This newsletter will introduce you to some new people and initiatives at UWB. We expect to see at least three new faculty come fall: Profs. Thamilarasu, Lagasse, and Lee. We have a new admissions advisor, Stacey Doran, who is helping us reach out to the community to encourage more students overall, and a greater diversity of students, to pursue studies to prepare themselves for computing careers. There is substantial information about our new MS in Cyber Security Engineering degree — we are very excited about it! We have been developing a variety of innovative approaches to computer science and software engineering pedagogy that combine technical fundamentals with professional development; Valentin Razmov’s course on developing modern software is just one example of this.

As always, this issue includes a focus article on a faculty member’s research — this time, on Munehiro Fukuda’s work on parallel computing for spatial simulations. The fact that we can have such an article in every issue, twice a year, for many years, and yet we still haven’t exhausted our faculty members’ research activities speaks to the breadth and quality of the scholarship in UWB CSS. As the Interim Chair of this new (yet old) Division, I truly feel that it is an honor and privilege to work with such excellent colleagues.

Cordially,

Dr. Michael Stiber
Professor and Interim Chair
Why should UW Bothell have a Master of Science in Cyber Security Engineering?

A common policy of cybersecurity is to respond to an issue that has already occurred. It is only after the attack has happened, and is apparent, that any action is taken, which is why detection is a major problem with many organizations. They often don’t know or don’t understand the vulnerabilities of their system and have no way of preventing an attack or even realizing if one has happened.

That is why teaching avoidance is so important. If you can understand the vulnerabilities of your software, then you can begin to predict the most likely ways it will be attacked and can prepare a system to detect an attack if it occurs. Students with this degree will know how to avoid the issues as opposed to just knowing how to respond to one already in progress.

How is this degree different from others?

The Master of Science in Cyber Security Engineering places focus on software engineers. Other cyber security degree programs are oriented towards software managers and system administrators, shifting the focus to the business aspect of the spectrum and not dealing with the software development lifecycle. This can lead to a lesser understanding of the vulnerabilities of the software produced, relying more on others to fix a security threat after it has been exploited. This degree will teach students how to be able to detect both an attack from an outside source or an abuse of permissions internally.

What types of courses can students expect to take?

The degree will consist of three types of core courses that will prepare graduates for a quickly evolving field that stresses privacy and security:

First information assurance is key in developing a strong background in cyber security. Important concepts of risk and hazard management will help students appreciate that avoiding a breach of security is far preferable to responding to one.

Next the secure development lifecycle is a spiral model software development process that consists of security practices to help students develop software that is resistant to attack.

Finally courses focusing on technical skills implementing cyber security and tools to detect secure network applied cryptography.

What career opportunities can students expect to find with this degree?

Graduates can excel as both analysts or engineers and a wide range of other career paths that require a strong background in software engineering and secure development lifecycle training. The degree also prepares students for CSSLP (Certified Secure Software Lifecycle Professional) certification and will allow many to go on to become leaders in formulating security policy.

Graduates will also find that they have a large number of diverse specified projects, done individually or as part of a small group (of 2-4 students) – had taught them how to tackle challenges of a more limited scope by focusing mostly on the implementation and (to a limited extent) the design of software, while assuming away all else.

CSS 490 aimed to expand on all these aspects and more in the interest of closing the gap on what a project experience feels like “in the real-world” – much longer, open-ended projects with ambiguities, done as part of medium-size teams (of 6-8 students), and involving requirements elicitation and testing on top of full-scale design and implementation… plus any challenges bound to arise from driving a project themselves and interacting with several teammates as part of the project work. It is worth noting that the goal of offering a “real world” project experience is dictated in no small part by many students’ self-professed aspiration to experience what a “real project” feels like, and their need to build their confidence in doing such work prior to finishing school and diving in head-first.
SEBASTIAN BLISSETT
The Boeing Company

Sebastian was entrusted with the task of securing communications for a client/server application at Boeing. As part of his 3 month internship, he gained knowledge in high level security standards, as well as learning the best security coding practices when working on designing and implementing software.

KACEY CADHOF-HOLSTON
PGXExperts

Kacey worked as a Software Engineer at PGXExperts building a PostgreSQL backed application which paired with the Hydra web app that collected and stored data on fish for distributed fisheries researchers using acoustic telemetry. This was a wonderful experience for Kacey as she worked on a real world project and gained an abundance of skill and knowledge and was hired full time by PGXExperts.

ALEXANDER ABBOTT
Cybersecurity Scoring Engine

Alex created a scoring engine which the UWB Cybersecurity team used during their practices. The scoring engine programmed in Python utilizing object-oriented program methodologies, multithreaded design, and a pluggable architecture, with a web interface for managing the engine and its configuration. The goal of the project was to monitor uptime of several services on one or more machines for multiple team users during control periods of time in order to facilitate the team in improving their skills.

STILL HAVEN’T STARTED YOUR 497?

DON’T WAIT, IT’S EASY!

Come in to the CSS office in UW1-360 and schedule an appointment with our internship coordinator, Janet McDaniel. Work to build a strong resume, look for an internship, and experience mock technical interviews to help prepare you for a job after school.
DEREK MCLEAN

**BrainGrid**

BrainGrid is an ongoing research project into efficiently simulating neural networks with general purpose graphics cards (GPGPUs). Such simulation aim to understand how brains, particularly human brains, work. Derek, in collaboration with Dr. Michael Stiber, Hugo Ribiero, Paul Bunn, and Christopher Burgess, implemented a new architecture for Brain Grid. The new architecture allows developers to easily create their own efficient neural network simulators on GPGPUs. Continuing on this work, the project is currently readying the simulator for an open source launch in summer 2013.

KATHLEEN HARWOOD

**Spacelabs Healthcare**

Kathleen work at the Verification department of Spacelabs Healthcare as one of the two people doing automation testing. After a crash course warm-up project using languages and patterns she had never heard of and playing with Windows Automation, she was put to the task of making a failing application usable by the automation team. This application had threading issues, confusing interface, and did not fulfill the specifications originally set about during development. Armed with the foundations of her education and some key usability classes, she set about making the application functional and usable to fulfill its intended purpose.

DANIEL STEWART

**26Juliet LLC**

Working at 26Juliet provided a fantastic opportunity to utilize all the classroom knowledge gained in a workplace setting. He was entrusted with all the technical responsibilities of the project, from design to implementation to debugging. The technical skills Daniel learned at UWB helped provide a strong foundation on which he could contribute to this company immediately, as well as grasp new and important skills quickly. Courses such as Management Principles for Computing Professionals & Analysis and Design provided valuable skills in working closely with customers. Working at 26Juliet also revealed just how valuable each and every single course within the CSSE program truly is.

WAYNE PADCAYN

**High 5 Software**

Wayne had a great internship experience at High 5 Software. He worked on a new application that provides solutions in managing a service business. He conducted usability testing to gather data and user feedback. He also did software testing to ensure the quality of the application, made improvements to the overall user interface and navigation based on analyzing results, as well as designing new modules to add to the application. He also created reports and key performance indicators for the application dashboard while collaborating with developers from different cities in Washington and out of the country.
The course therefore offered opportunities for growth and satisfaction to students with diverse sets of interests and different levels of prior experience. The ideal goal was that those who choose to actively engage would come out with a strengthened confidence that what they might encounter in a future workplace setting would already be familiar, well practiced, and well understood.

The instructor had used a similar structure in CSS 360, with success, and indeed many of the enrolled students had taken his CSS 360 (and/or CSS 370) course before, and now wanted even more of it: hoping to re-enact the same inspiring atmosphere while diving deeper into practicing software engineering “from womb to tomb”.

This expanded structure was intended to create a sense of wholeness to the experience – a deeper understanding of how the various "parts" of software projects tie together organically, and what it takes to move a project’s development from start to finish through the different phases.

Students owned the decisions on virtually all project aspects, and rightfully felt in the driver’s seat. They were encouraged to conceive and propose “half-baked” project ideas, on which everyone voted their preferences, so the proposals with the highest support were selected for further development by teams of students who had expressed strong motivation toward those ideas. From then on, the focus was on teams moving forward on their projects, aided by regular, individualized coaching by the instructor. The instructor intentionally refrained from forcing any decisions on the teams, so as to let students own their experience and their learning from it; otherwise, the tendency is to defer to the instructor’s judgement and later on to disown the experience and disallow the learning.

How did it turn out?

By the end of the quarter, all teams had brought to life their initial half-baked ideas and demoed working prototypes, many of which were impressive achievements considering how little most students knew just a few weeks earlier about their subject domains, their necessary technical tool sets, and their fellow teammates.

All teams had experienced unique challenges and joys along the way, learning relevant lessons and, above all, feeling that the course was time well invested on their part.

Most teams and individual students had adjusted their goals (team stretch goals or personal stretch goals) as a result of what they had learned during the project development. They understood the tradeoffs involved in project work on a deeper, more practical level.

Finally, a few individuals expressed intent to continue developing their projects beyond the confines of the course. This is a delightful expression of their personal investment, a testimony to the meaningfulness of the project experience for them, and an appreciation of the value of what was already created.

What is behind the success?

These results reflect several core teaching principles, espoused by the instructor and applied in courses beyond CSS 490:

-upholding high student motivation by incorporating aspects around which students have passions (e.g., student-proposed projects of interest, meaningful technical work, pursuing personal stretch goals), while removing motivation sinks (e.g., inflexible mandated schedules, focus on grades, stress) as much as possible;

•personalizing the experience to the unique interests and needs of the students and project teams, which makes each version of a course a unique and creative experience (e.g., by spending time to coach project teams individually and regularly, by offering mini-lectures on relevant topics of specific interest to students in the class);

•flexibly adjusting aspects of the course (e.g., schedule of deliverables, curriculum depth and ordering) in order to optimize the effect of the above two principles (e.g., the agile approach applied to teaching);

•shrinking the distance usually perceived by students between instructor and themselves (e.g., by consulting with students regarding decisions that affect the course, their teams, or themselves individually; by the instructor acting as a coach and a consultant, but not a decision maker on students’ projects); and

•bringing pragmatism to the classroom (e.g., by creating the space for realistic projects to conceive and flourish, by steering toward project themes that students can be proud to put on their résumés), so that students can rightfully feel that they have acquired skills that are both marketable and transferable across disciplines, which in turn would boost their confidence in becoming successful in their future careers and in life.
environments to work in. Many large software development companies need specialists in security and privacy and the technology associated with it. There is also work in specialty organizations that work with larger organizations in security review and penetration testing as well as dealing with direct threats and responding to discovered vulnerabilities.

In this region, many biotech and healthcare device manufacturers will need graduates who can create software that is reliable and secure. These, along with many large corporations and countless start-ups, will provide a huge advantage to our students.

What projects will students work on in this degree?
Students will explore actual cases, being able to experience real problems in industry and reanalyze decisions both good and bad. They will encounter mock applications for them to determine security threats and live virtualized environments to be able to practice detecting the difference between vulnerable and exploited. Additionally they will assume technical roles of engineers and be able to react and respond to various threats.

What were some of the biggest challenges you faced developing this degree?
An important, yet difficult, task when developing the program was coming up with a complete definition for the curriculum, as the degree is very unique and is not being duplicated anywhere else. There are many challenges associated with requirements in the field of cyber security, and determining what prepares someone to not only enter but succeed in the environment was an enormous task to face.

The degree also has more pre-requisites than others and mapping the degree to government defined needs in the field of cybersecurity required a balancing act between what a student should already know and what they should learn in the program to optimize their success.

‘FOCUS’ CONT.

portion of the network is useful, so this could be done in parallel. Prof. Eric Salathe from Science & Technology is working on climate change research. He works with very big data and analyzes weather simulations. He needs to find change in the data to determine what factor causes the change and how this change will last and will it change the future weather. This work is being considered along with professor Asuncion who is interested in data provenance search. With this we would like to apply multiple agents to this big data, which is stored in a 3-D array, so we can deploy many agents to this big data and will try to find out the change of the data and determine cause and effect.

Another interesting application I have worked on previously was orchard temperature prediction. In the period from April to June, apple orchards are affected by frost caused by sudden temperature drops early in the morning. When this occurs, farmers turn on sprinklers because the water actually emits heat, so by saturating the ground the air temperature increases. Concurrently, the farmers turn on wind generators. Generally cold air moves in a stream just above the ground going up until twenty feet where the air becomes warmer. These generators are used to steer the warm air downward closer to the orchard. Both the sprinkler and wind generators are used to combat morning frost. The problem is the exact timing and exact location to turn on these systems, because they are expensive to run and farmers cannot afford to have them on all day. So a farmer has twenty temperature sensors over a one hundred twenty acre orchard monitoring the temperature transition overnight, especially on calm cold nights. The question is, how can we predict the temperature transition? This could be done through computation fluid dynamics and using MASS library so we can parallelize this data and the temperature predicting application.

So this is kind of the work we are focusing on. So far three students have completed their masters theses. Two of them worked on a Java based MASS library and a GPU based MASS library. The other has worked on a computer job management program. So the key idea is while we have MASS described application, the problem is on which processors would we like to run this program? This student wrote it so that the application will be automatically dispatched to the multiple computing nodes.
Help Support Our Mission!

Whether it’s $5 or $500, every gift helps keep the Computing & Software Systems program a unique and rewarding experience for our students and an important resource for computing education in this region.

Support from our alumni and friends enables us to grow our activities even in these tough economic times.

Become part of our mission by donating today. Direct your giving to CSS by visiting www.uwb.edu/css and clicking on the ‘Make a Gift’ link.

Thank You for Your Generous Support!