



Programming Systems Laboratory  
Department of Computer Science  
1214 Amsterdam Avenue, Mail Code 0401  
New York, NY 10027  
212-939-7081/fax:212-939-7084  
kaiser@cs.columbia.edu  
<http://www.cs.columbia.edu/~kaiser>

January 17, 2012

To Whom It May Concern:

I am pleased to very enthusiastically recommend Mr. Jonathan Bell for a student volunteer position. Jon is my newest PhD advisee, just entered the PhD program a year ago, but I have known him for 3 1/2 years.

I met Jon when he took my fall 2008 course W4156 Advanced Software Engineering, as the first junior I'd ever encountered in that class – which I first taught in 1989. Jon was one of the handful of standout students in the class, out of 60 that term, and received an A+ grade. W4156 is part of the MS 'core', so the enrollment is almost entirely graduate students. After the great experience with Jon in the class, I've gone to some effort to try to attract more undergraduates (who can take it as an elective).

Jon also took my spring 2009 course, E6125 Web-enhanced Information Management while still a junior, also the first junior I'd seen in the course - in that case dating back to 1997. He likewise received an A+ even though 6125 is an advanced graduate seminar.

I went to extra trouble to arrange for Jon to be a TA for the software engineering course in fall 2009; the 'trouble' was because we do not normally assign undergraduates (he was then a senior) to TA graduate courses, particularly not those that involve mentoring team projects. But, as I expected, Jon did a superb job – with rave reviews from the student evaluations. Jon also did great work TAing the same course in fall 2010 and 2011. Last year he presented one of my lectures with only about half an hour notice when all routes from northern NJ to Manhattan were blocked off by the police for several hours (so I could not get to campus), and both terms he gave several pre-planned lectures (that I attended) that went very well.

Although I initially met Jon in the context of teaching, his most important strengths lie in research – particularly interdisciplinary research, so far working primarily with computational biologists but we're starting to interact with educational technologists and social scientists as noted below.

Starting in summer 2009, Jon completed several 'independent study' research projects in my lab. The first one concerned performance studies of *CPU Torrent*, a facility previously developed in my lab for sharing/offloading CPU cycles somewhat analogous to BIT Torrent's bandwidth sharing/offloading. Many computationally demanding scientific computing systems consist of a 'proprietary' subjob that can only execute in the environment of a single provider together with one or more 'utility' subjobs available from numerous providers. The burden of supporting external community use of the software system tends to be very expensive, resource-intensive, and leads to very long wait queues (often over 24 hours) if the proprietary provider must also supply the utility subsystem – as is at present the norm.

The immediate application was to BLAST, a utility component of many CPU-intensive genomics analysis tools. The gist is that the burden on the proprietary provider as well as external users, typically NIH-funded academic researchers, can be minimized by separating the BLAST subjob and running it at government data centers or NIH-funded computation centers. While this sounds simple, the technical challenges involve distributed computing concerns such as unreachability or other failures

regarding the utility providers, as well as dynamic reaction to the cross-over point regarding when it is better to run locally rather than offload. The system architecture is not in any way peculiar to BLAST or computational biology, but of course some implementation details as well as the measurements had to be.

In his later projects, Jon developed several key 'collaborative work' improvements to geWorkbench, a plugin genomics platform developed by Columbia's Center for Computational Biology and Bioinformatics (C2B2) and disseminated by NCI. The gist of our *genSpace* plugin is to continuously data-mine 'workflows' from the activities of previous users, to enable in situ suggestions to current users regarding what to do next as well as graphical depictions as to how the user's known tools fit into frequently used tool sequences. This is helpful since most new geWorkbench users have prior experience with only one or two geWorkbench tools, which they previously used stand-alone, and the learning curve to productively interact with the 70 or more tools available through geWorkbench has been found quite daunting. Manuals, tutorials, etc. are labor intensive to produce and maintain, and users don't read them anyway, whereas *genSpace* recommendations are always up to date (as in popular "people like you..." collaborative filtering applications as used by netflix, amazon, etc.). I would not normally involve an undergraduate in presenting my lab's research to C2B2's executive director and, later, NIH/NCI representatives at the C2B2 annual retreat (spring 2010), but did not hesitate in Jon's case – and he did an amazing job throughout the series of demos and poster presentations.

Although Jon intended to pursue a terminal MS when he enrolled in graduate studies in fall 2010, I convinced him to 'transfer' to the PhD program after one semester. Besides continuing work with *genSpace*, Jon is the lead student on our new *HALO* (Highly Addictive, socialLy Optimized) project, where we employ MMORPG motifs to engage software engineers in the more tedious aspects of the software process, particularly testing. Software engineering tasks are presented as solo and group 'quests', rewarding experience points leading to 'leveling up'; conscientious developers can earn titles like 'The BugSlayer'. *HALO* is based on ideas from learning scientists (we're collaborating with a lab at Teachers College) to leverage operant conditioning, flow, etc. as part of alternate reality games for constructive purposes. We have already published one workshop paper with Jon as the first author and he recently submitted a conference paper.

Jon is extremely smart and extremely articulate, both of which are critical factors in successfully conducting interdisciplinary research – which as I see as a primary milieu for future software innovation. Overall, I think Jon would be a wonderful choice for one of the student volunteers. His English proficiency (native English speaker), willingness to help others and organizational skills are all exemplary. Do not hesitate to contact me if additional information is needed.

Sincerely,



Gail E. Kaiser  
Professor of Computer Science  
Director of the Programming Systems Lab  
Chair of the Doctoral Program  
Associate Chair of the Department of Computer Science