Educating System Administrators

CHARLES BORDER AND KYRRE BEGNUM

Charles Border teaches networking and system administration classes in the BS and MS programs in networking and system administration in the Department of Information Science and Technology at the Rochester Institute of Technology in Rochester, NY. He is an active international educator offering classes at RIT Dubai and in partner programs in the Dominican Republic. His areas of research include system administration education, international higher education, enhancing the innovation and creativity in professional master's degrees, cloud computing, and teaching in the cloud. Mr. Border earned a PhD in higher education administration in 2000 and an MBA in 1993, both from the University at Buffalo. He lives in South Wales, NY, with his wife Kathleen and their three children.
cbbics@rit.edu

Kyrre Begnum, PhD is an associate professor at Oslo and Akershus University College of Applied Sciences in the Department of Computer Science, and is an adjunct professor at Gjøvik University College in Oslo, Norway, where he teaches primarily systems-related courses. Dr. Begnum has published extensively in the field of cloud computing and virtualization.
Kyrre.Begnum (at) hioa.no

If you are a long-time attendee of LISA conferences, you will be very familiar with the educators groups that have met either as part of LISA or just before the LISA conference. The Summit for Educators in System Administration (SESA) had its first official meeting in 2013 under the guidance of Kyrre Begnum from Oslo University College and Caroline Rowland from the USENIX Board and NIST. The meeting was a big success with more than 30 academics and others interested in the education of the next generation of system administrators in attendance. Later in 2014, the USENIX Board decided to embrace SESA as a new group under the USENIX banner.

During meetings around SESA, we decided to petition the USENIX Board to form a journal, separate from but affiliated with SESA, called the *Journal of Education in System Administration (JESA)*. Our vision for SESA and *JESA* is to give academics and others interested in the education of the next generation of system administrators a place to discuss their efforts and to share best practices.

The reason it makes sense to do this under the USENIX banner, rather than the other academic computing organizations such as the ACM or the IEEE, relates to our vision of system administration and operations as a very applied field within computing that has not received its fair share of respect within the more theoretical computing organizations. As academics, we feel more at home in the USENIX community and feel that it is a better home for our vision of what system administrators do in the world of work. The professionalism of the USENIX community fits better with our vision of what we want to instill in our students, and we look forward to working with the community to help us advance our shared profession.

The Future of Computing

Computing, as an academic discipline, has just hit its fiftieth birthday and is undergoing a period of introspection something like what many of us go through around mid-life. Since its inception, the idea that computer science was really a "science" has been an item for debate [1]. One of the main concepts behind the idea that computer science was not a "science" was the notion that, unlike the other three branches of science (physical, life, and social sciences), computer science dealt with an "artificial" environment. This is becoming less persuasive as we start to gain a better understanding of the similarities between the computation that we do with computers and the computation involved with some of the most basic life processes such as evolution, natural selection, chemistry, gene regulatory systems, and neuronal networks [2].

There is another way to look at computing as "the union of three disparate perspectives: the mathematical perspective, the scientific perspective, and the engineering perspective." [2] From this perspective, computing as a discipline derives its use of various formalisms from mathematics, its drive for continuous improvements from engineering, and its desire to make empirical predictions from the small and simple to the very large and complex from all the sciences.

As computing has evolved from its emphasis on engineering and the initial development of systems in the early days to our current situation of very large and complex systems, the question still remains: What is the future of computing? Frustrated with all the other names (e.g., the science of computation, or the science of computer science), Richard Snodgrass has proposed an entirely new term to describe the future of computing, ergalics. "The goal of Ergalics is to express and test scientific theories of computational tools and of computation itself and thus to uncover general theories and laws that govern the behavior of these tools in various contexts. The challenge now before the CS discipline is to broaden its reliance upon the mathematical and engineering perspectives and to embrace the scientific perspective." [3]

This matters for us as we begin to think of working to define curricula around system administration, because we need to have a better understanding of what the goals and outcomes should be for our programs. One of the biggest changes that has happened to higher education in the last decade or two has been a growing demand from our stakeholders for an increase in accountability for the resources that we consume. This has been instantiated through the rise of the assessment movement.

Measuring the Performance

Assessment requires that each program approach the measurement of the success of the program from a three-step process. Each institution defines a mission statement that defines who the institution serves and the relationship of the institution to the world around it. Based on this mission statement, each program defines a set of broad program educational outcomes that define the characteristics of the graduates of the program three to five years after graduation. The idea behind having this time lag is that we do not want to educate our students to just be able to get that first job; we feel that educating students properly prepares them to be lifelong learners. Measurement of a program's educational outcomes is a problematic thing. A lot of important things can happen to a person between the ages of 22 and 27, and just contacting our graduates can be difficult. Our attempts to measure program educational outcomes are imperfect at best and rely on the use of surveys and our other contacts with our graduates. Lastly, each program defines its student outcomes, which are the things that students should be able to do when they graduate. Through our assessment process we measure the results of our programs, and we feed back the results of our assessment into program changes to make the program better over time.

Student outcomes are generally measured in individual courses that all students are required to take. The actual means of measurement depends on the type of outcome. If an outcome relates to a student's ability to communicate effectively, we might measure this by grading student writing assignments against a rubric that breaks down the grade for the assignment into several categories, with a score assigned to each category. If the outcome relates to the ability of a student to do something (e.g., configure a BIND DNS server), we might have a standard lab assignment that all our students need to complete that is graded against a rubric.

In the old days prior to assessment, we asked our constituencies to trust us about how good a job we were doing. Now we have a process in place to measure the contribution each program makes toward the institution's ability to live up to its mission statement and find ways to make the program better over time. From the perspective of a teaching faculty member, a couple of the most important points about this process are that we develop our own set of program educational outcomes, student outcomes, and a process by which we use metrics to improve the program. This may sound very bureaucratic, but in the end it is very much a faculty developed and led process through which we can improve our programs.

From a day-to-day perspective, assessment, in essence, asks us to define a set of program goals, break those goals down into outcomes, align those outcomes with specific courses, and find ways to measure the ability of each course to contribute to the overall success of the program. If a course does not enhance the ability of students to satisfy the program outcomes, it should be removed from the curriculum. If students successfully accomplish the goals we have set out for them, we acknowledge that and move on, and if they do not, we examine what we do and try to find ways to do a better job in the individual courses.

To make assessment work effectively, we need to have the right program outcomes, and this is the area where a curriculum that concentrates on the applied skills of a professional system administrator should be very different from the more theoretical skills of a computer science program. If system administration and computer science program outcomes were the same, there would be no reason to have a separate system administration program.

Program Outcomes

There are two dimensions to the design of new program outcomes. The first dimension concerns the window of time that students are in our programs. What can we teach students when they are fresh out of high school that will be meaningful to a career that does not begin for four years? A relevant way to think about this is to reflect on the systems practices that we were pursuing four years ago. What has changed since then and what has stayed the same? A related issue is the amount of time that we have with the students. If we are to add things to an already busy curriculum, what can we take out? If we add configuration management because we decide that every systems person should have a working knowledge of how to maintain the consistency of the configuration of many machines, can we remove a semester of Java? Tradeoffs must be made, and the place that we make them is in the program outcomes.

A second dimension of the development of program outcomes relates to what we determine are the main things that you want an entry-level system administrator to be able to do. Is programming in a specific language such as Perl or Python the most important thing, or is it more important to have a general understanding of, for example, troubleshooting skills or service deployment architectures? When we work to develop the educational outcomes associated with programs, we have to be very careful, because you might get what you ask for. If we lean too heavily on the needs of the moment will we have a very narrowly skilled employee who is unable to adapt to changes in the demands placed on him? Even though many organizations are relying heavily on Puppet for configuration management and EMC for large scale storage architectures, should we design our curriculum around these specific technologies or should we concentrate on developing a more generalist curriculum that stresses things like Bash scripting, Web services, and storage area networks and networking?

There is a difference between developing lab exercises that require students to use specific technologies (deploy this Web service on this Web server, running on this operating system) and building context around basic technologies by discussing the general concepts involved with the technologies. The concepts last, but the specific technologies change very rapidly. The same distinction applies when we develop the outcomes associated with our programs. If our outcomes are too specific and technology-focused, we run the risk of having to change our outcomes with each iteration in technology, and of having students whose education loses its relevance before they even graduate.

Student Recruitment

We have heard from many of you that it is very difficult to recruit the right new employees for your businesses, and we in higher education have heard you and we want to help. But we also have a problem. The kind of very bright, hardworking, and creative students that you want to recruit to run your systems have many options when they choose a major and very little understanding of what the different majors and the careers they lead to actually consist of once they graduate. Many students show up at college not knowing about different careers, but knowing that they want to major in something related to computing. While this is fine, it presents a problem for those of us seeking to recruit them into a specialized field such as system administration that they may never have even heard of. Although this generation of students is just as rebellious as we were (which is good), parents play a larger role in the student's decision-making than we usually give them credit for. But the same problem remains: The parents may not know what a system administrator is either. The current growth in computer science enrollments may be a response to the uncertainty that many people feel about jobs (let alone careers) these days, with students opting to major in the more well known, generalist computer science degree rather than a specific career path that they don't understand and that might (so they fear) be outsourced, leaving them in debt and unemployed.

This is particularly a problem as we try to recruit a more diverse student body. Just as industry is being asked more pointed questions about the diversity of their employees, we are also receiving the same types of questions. It is very important for us to expand the pool of students interested in systems, educate all the students, and create an environment where all students can succeed.

To successfully recruit the kind of students that you will want to recruit as employees, we need to create an interesting curriculum that allows students to gain an understanding of the field of system administration and, at the same time, excites their interest, creativity, and problem-solving skills. Our goal in developing a system administration curriculum should be to develop our students into employees who feel empowered to be creative and find their work engaging, interesting, and worth concentrating their efforts on.

The Future of System Administration and Operations Education

With only a very few exceptions, computing programs in higher education are dominated by computer science programs based on a more theoretical understanding of what computing is all about. While this might be sufficient for many careers in computing we don't feel that it is the right approach for all careers in computing and all organizations. The goal of SESA and *JESA* is to create a venue where people interested in a different side of computing can exchange ideas and information relevant to the development of new curricula in system administration. These new curricula may come in many different flavors, with some being more programming focused and others more focused on hardware/service deployment issues. And they may rely on different phrases to describe their curriculum (operations seems to be a bigger concept than system administration, but if students don't know what system administration means, they certainly don't know what operations means) and/or rely more on business concepts (operations management is an interesting topic to many people) than strictly on computing, but we want to provide a place for all of them.

For the academics reading this article, we want to provide a place to discuss your plans for the future and goals for your curriculum. For the industry people reading this, we want to encourage you to become involved both in our new group and especially with your local colleges and universities. They need your input into the curriculum design process, and they need your talents as an instructor. If you have never taught a college course, you may find it to be a very interesting change of pace for you that puts you in contact with some very bright and hard-working students and gives the students an opportunity to benefit from your experience. At SESA we also need your thoughts and experience as we try to distill from our rapidly changing industry those things that will last and that can form the basis for an interesting and challenging curriculum.

References

[1] Peter J. Denning, "The Science in Computer Science," *Communications of the ACM*, vol. 56, no. 5 (May 2013), pp. 35–38.

[2] Richard Snodgrass and Peter Denning, "The Science of Computer Science: Closing Statement," Ubiquity Symposium: The Science of Computer Science (June 2014), DOI=10.1145/2633608.

[3] Richard Snodgrass, *Ergalics: A Natural Science of Computation* (University of Arizona, 2010).