The Doctor of Professional Studies (DPS) in Computing at Pace University provides computing and information technology (IT) professionals a unique opportunity to pursue a doctoral degree while continuing to work full time. It supports interdisciplinary study among computing areas and applied research in one or more of them, and thereby provides a background highly valued by industry. It is an innovative post-master’s doctoral program structured to meet the needs of the practicing computing professional. The DPS in Computing, while advanced in content and rigorous in its demands, is distinguished from the Doctor of Philosophy (PhD) by focusing upon the advancement of the practice of computing through applied research and development. It is designed specifically for people who want to do research in an industrial setting.

The DPS program provides an intellectually stimulating learning environment in which emerging computing and information technology can be discussed and researched in an open forum. Students and faculty are encouraged to share their experiences and ideas with everyone in the program.

While a PhD advances knowledge in the discipline of study and a professional doctorate advances the practice, in computing there can be a fuzzy distinction between the two concepts. Successful professional doctoral students will demonstrate that they have effectuated the practice as well as effectuated deeper and broader understanding of the practice through research. Our students regularly have the opportunity to put their learning into practice. This is particularly true because they hold senior positions and are able to effect change in their organizations. There is value added by practicing computing and studying at the same time.

The DPS in Computing admitted the first class of twenty students in the fall of 1999. To qualify for admission, students complete a master’s degree and have a minimum of five year’s professional experience in computing and information technology, although typically our students have 10 to 20 years of experience. Each class brings professional expertise in specialized areas of computing to the learning community. The class proceeds through the program as a cohort, and is expected to graduate together after three years.

The program is unique in its design and methodology. The motivation for the program, its structure, and early experiences were described in a paper written as the program was ending its second year, when the initial class had completed its coursework and was embarking upon the dissertation.[13] Another paper, focusing on assessment, written at the end of the fifth year[14] looked at how the dissertations compared to doctoral dissertations from traditional programs, considering the significance of the problems studied, the investigative methodology, readability and the dissertation completion rate.

The DPS in Computing is now in its eleventh year. Through the class of 2008 (the cohort who matriculated in 2005), 52% of the 118 students who entered the program, have graduated. The average time in the program for the 62 graduates is three years and two months between matriculation and the successful dissertation defense. The shortest completion time is two years and six months, and the longest, five years and ten months.

What kind of doctorate to offer?
It is useful to make the distinction between a professional doctorate in computing and a research doctorate for computing professionals. The professional doctorate is often perceived as different from and not a substitute for a research doctorate, most often, the PhD. There has been increasing interest in professional doctoral programs.[6, 15] But reports cited indicate that there is confusion about what is desirable in a professional doctorate. Should there be a research dissertation requirement as in the PhD, or should there be a significant contribution to practice requiring some demonstrated practical action that produces change or development in a community of practice or in an organization? We set our doctoral program apart from both the professional doctorates and from the academic research doctorate (PhD). It is neither one nor the other, but a synergistic integration of the two.

We had to choose what the degree designation would be -- PhD, DPS, or some new designation, e.g., Doctor of Computing. The last choice was ruled out because the New York State Education Department does not recognize such a designation. Pace University pioneered the professional research doctorate when it introduced the Doctor of Professional Studies (DPS) in Business in the early 1970s. In 1999 we launched the DPS in Computing, a doctoral program that integrates computing and professional cultures. The DPS has been considered by the National Science Foundation “to be a research doctorate equivalent to the PhD.”[16, 17]
The profession of computing
At the time the DPS in computing was in development in 1998, the notion of computing as a profession was garnering interest. Peter Denning published his essay about the nature of the emerging “Profession of Computing” and its relation to the computing disciplines -- computer science, software engineering, software architecture, and other IT areas of specialization.[8] In addition to a shared body of knowledge and skills, a profession has a responsibility to the community -- its effect on the community is for the good -- and a professional must understand how the profession and the external community affect each other.[1,12] Peter Denning proposed an answer: “Who Are We?”[9]

The computing professionals who comprise our applicant pool are the set of people who work with computing technologies. But what are these computing technologies? We need to understand what we mean when we call the degree Computing. What should be in the curriculum? What are appropriate areas for dissertation research? What backgrounds do we look for when admitting students?

The academic view of computing is that it is primarily composed of the disciplines of Computer Science (CS), Information Systems (IS), and Software Engineering (SE), and further specialized by the research topics, approaches and methods found in each of these disciplines.[10, 11] Another perspective is that it is defined by industry computing needs and functionality. We embrace both views of computing as we characterize our program curriculum, philosophy and dissertation research areas.

In our domain, computing involves the interaction among computer-based methodologies with other disciplines so that we can:
• explore the many uses of computers;
• elicit meaningful problems/applications that can benefit from computer-based solutions;
• suggest unsolved computing research questions;
• study how people interact with computer-based applications and the effects;
• pursue research challenges coming from applications and other disciplines.

Addressing the research challenges from non-computing domains can involve computing research applied to those domains or applied to one of the computing disciplines. An important aspect of computing research is that it is interdisciplinary and may depend upon the synergy of more than one discipline, i.e., where neither computing nor another discipline alone can create new knowledge. Because of their professional backgrounds, our students tend to think about research in an interdisciplinary manner. This stimulates their ability to bring to light new problems that neither discipline alone would recognize, and that can often lead to fundamentally new ways of thinking about problems. This might include the implementation of known computing solutions to applications in new ways, or the development of new tools and techniques and theories to solve new problems or to solve known problems in better ways. The contribution might be to computing or to the practice in an application domain or both. Much of the research that our students do has immediate effect because they are working in and with other disciplines.

Our students
Our students have been educated in and are professionally involved in an extremely diverse set of computing areas and applications. The DPS in computing attracts and thrives on a professionally diverse student body. Entering students’ careers extend from 5 to 33 years, with an average among students of thirteen years. Professional activities include software development, project management, telecommunications and network design, education and training, IT management, data management, and Internet engineering with employers such as Verizon, IBM, MetLife, Dannon, USA, JPMorgan Chase, AT&T, E*Trade, Brookhaven National Labs, Oracle, Museum of Natural History, Pfizer, and Sothebys. Their job titles include CIO, CTO, vice president, director, consultant, senior technical staff, project manager, software architect, data administrator, professor. Many of our DPS students live and work in the greater New York metropolitan area, but others travel from Massachusetts, California, Illinois, Pennsylvania, Washington, DC, and Canada.

Our faculty
Unlike traditional doctoral programs with a focused area of interest, e.g., computer science, information systems, or data communications and networking, our doctoral faculty need to have broad multidisciplinary interests and experience in order to support and facilitate the spectrum of dissertation research topics. This works for us at Pace University because we have an extremely diverse and eclectic faculty. We are also fortunate to be able to reach out to external professionals and researchers to complement our own faculty as needed, typically from the students’ employer contacts and faculty colleagues.

Faculty time commitment may tend to be greater than what is normally expected in traditional doctoral programs; this is because our faculty act as research and learning facilitators in addition to being instructors and dissertation advisors. They are part of the teaching-learning community of support and actively interact with all the students, not just advisees. All members of the community -- students and faculty -- are proactive in supporting and encouraging the dissertation progress of every student in the class. This is particularly apparent in the doctoral symposium-like third-year class meetings.

The program
Each September, a new class of 15 to 20 students matriculates, and classes move through the program as a cohort. The program uses a team approach to both teaching and learning, and combines monthly face-to-face weekend meetings with asynchronous distance learning via the Internet.

A special strength of the program is the specialized technical knowledge and expertise embodied in the students’ professional computing and IT experience. A learning-teaching community is created at the very beginning which draws upon this expertise and strengthens the collaborative skills of the students.

Every indicator confirms that a high level of cohesion grows quickly and persists. In fact, this strong community may be
credited for the retention of students some of whom feel overwhelmed by the pressures of school and work.

We do not require additional computing and workspace resources because our students have their own laptops and except for the time they are onsite (once a month) they work in their own space.

**Curriculum content and delivery**

Because students come with different computing and professional backgrounds and experience, we do not assume specific prerequisite knowledge in the same way that we might, for example, in a CS degree. We do not use a compartmentalized course structure that is traditional in graduate education. We have some courses that fit that structure, but these, too, have an “open” component, i.e., team projects where teams select the specific area of study with faculty approval and guidance to enhance the total learning experience of the class.

The challenge is what to choose for the curriculum. We cannot assume common prerequisites in a class with diverse computing backgrounds – CS, IS, software engineering, telecommunication and networking, MIS, various business or social science areas. The program is about computing.

We chose a program in “computing” as different from CS, or IS, or any other specific computer related discipline for a doctorate serving computing professionals. A specifically focused program is not as relevant to seasoned computing professionals who already understand that computing is an integrated (synergistic) discipline. Students come from various and widely diverse application areas – health, education, business, engineering, data management. They appreciate how the ideas, principles and practices that they learn can be used among many applications, sometimes in the same way and sometimes in different ways.

One chief educational objective is that practicing information technology professionals will encounter and develop insights into recent developments within the fields contributing to applied computing; and they do so, in an educational and research setting. One goal is that students will be able to apply some of what they are learning immediately on their job. Sometimes the utility is concrete; other times it is more abstract, for example, in the way they think about a problem. The other chief objective is that each student will become able to undertake the creation of new knowledge and to report research results to academic and professional audiences. Rather than lecture, we more often use Socratic challenges. Students read, write and discuss while faculty (and eventually the students) challenge assumptions and conclusions. Throughout we posit that the process is more important than the product, at least in the educational arena.

We start close to the frontier of computing knowledge and practice and look back to understand how we got there, as opposed to starting at the beginning and trying to get to the frontier. Because of the diversity in academic and professional experience, every student’s distance to the frontier is different, and we depend on the teaching-learning community to provide some normalization. For example, one of the software development areas that we have been studying and experimenting with is dispersed and distributed agile methodology. A student working at a major corporate research facility has been exploring these emerging software development methodologies and practices, and was able to share some experiences with the DPS community.

**First year of study**

For the first year, we chose three major areas of study that every computing professional should know in professional and academic experience – software, computer communication and networking, and the Internet.

It is in the software curriculum area that we are the most adventurous and unique, including such topics as modern software development methodologies, patterns, cyber ethics, retrospectives. There is a strong focus on agility, with an emphasis on first being agile, and then doing agile. The students read the seminal authors, e.g., Beck, Cockburn, Highsmith, Coplien, Cohn, Poppendieck, and meet some who are invited to spend a weekend with the class. There is an agile software development project in which the students work in dispersed (virtual) agile teams. We view agility in a broad sense, based on a set of fundamental principles, not limited to software development. Of all the areas studied, the focus on agility has had the greatest impact on dissertation research as well as on effecting change in work environments. Agility, perhaps above all, characterizes our program.

A second area in which we are innovative is the study of patterns. Our students study and use software design patterns in the agile project. They also study organizational patterns and pedagogical patterns. Our students have been able to make changes in their professional work by using organizational patterns and patterns for introducing change.

A pattern provides a structurally simple way of describing an expert solution to a recurring problem. It captures an “essence” of a context, problem and solution. Patterns can provide a better means of understanding and communicating. Software and Organizational patterns have become an important driving force in the industry, transforming both software production and the organizations that produce it. Learning how to mine and write patterns helps the student to recognize and document the essence of their research.

In our program, students are encouraged to think and write holistically. We introduce them to “object think”, not to become object oriented programmers, but to change the way they think about things and the way things interoperate. Thinking in terms of patterns, and in terms of objects, have commonalities. Jim Coplien said it this way[4]:

*I like to relate the definition of patterns to dress patterns. I could tell you how to make a dress by specifying the route of a scissors through a piece of cloth in terms of angles and lengths of cut. Or, I could give you a pattern. Reading the specification, you would have no idea what was being built or if you had built the right thing when you were finished. The pattern foreshadows the product: it is the*
A pattern involves a general description of a recurring solution to a recurring problem replete with goals and constraints. But a pattern does more than identify a solution; it explains why the solution is needed.[3]

A “retrospective” is typically not included in a computing degree program. While a project retrospective is an important component of modern software development, we utilize the practice more generally. It has proven to be an area that many of our students have successfully introduced into their professional work. A retrospective helps facilitate process improvement and enables team learning. Since our program is heavily dependent on team and community, we hold periodic retrospectives of the program itself with the students.

Other software related topics studied in the first year include open source, ethics and technology, privacy-security, and the impact of tools on software development.

Second year of study
During the second year, we explore cutting edge issues in emerging information technology. The following are examples of what we have studied. Depending upon the interest and importance to the DPS community at the time, some of these topics constitute an entire semester course, and others are modules in Topics in Emerging Information Technology courses.

- Small computing devices – pen computing, handwriting and speech recognition
- Data security and information assurance
- Artificial intelligence and genetic algorithms
- Pattern recognition – visual patterns, speech, biometrics
- Internet performance and high-volume Web serving technologies
- Data mining, data warehousing and data modeling
- Pervasive computing
- Distributed components, middleware, Web services
- Patterns – software, organizational

Research seminar sequence
The purpose of the Research Seminars is to prepare the students for doctoral research. The seminars are what enable completion of the degree in 3 years. The seminar sequence begins in the first year of study with an introduction to what research in computing is about, utilizing examples of different kinds of computing research and methodologies and presented by faculty and invited researchers. In the progression of seminars students investigate research areas; the goal is a dissertation proposal draft by the end of the six-semester seminar sequence.

By the end of seminar 3 (summer of year 1) students select a research area and have a draft of an idea paper. This is a working document that grows to include the research approach and plan, and eventually to the final dissertation. We recommend that students choose an area of research in which they have prior knowledge and expertise. This is possible because all DPS students are mature computing professionals with a minimum of five years of experience in industry. Most students of traditional doctoral programs matriculate directly from the BS or MS degree and must get “up to speed” in an area; our approach can save several years of effort.

In seminars 4 and 5 (fall and spring of year 2) we use the Socratic method, asking questions and challenging what is said. Initially this is managed by the faculty, but the students learn to question and challenge, and quickly become Socratic. They are exposed to new ideas and ways of thinking. They learn abstraction, and see patterns and models of inquiry and discovery that they can apply to their own ideas. Students connect with an advisor in the spring of the second year. The DPS student owns the research and is immune from committee or advisor changes.

The dissertation process
Unlike many other programs, Pace DPS students focus on research beginning in the first semester of study under the guidance of faculty advisors. As they progress through the program seminars, readings, and discussions, they are exposed to emerging issues in computing and information technology. In most cases, the research seminars lay a foundation for a dissertation or indirectly stimulate interest in an area that leads to a dissertation. The program has built-in coaching and mentoring by faculty advisors and, most importantly, by the students themselves.

The DPS dissertation is a rigorous, original, independent applied research product that may advance knowledge, improve professional practice, contribute to the understanding of computing. It is a disciplined and systematic inquiry for the purpose of discovery, establishing or extending the field of study. The dissertation must be of sufficient strength that the student is able to distill from it a paper worthy of publication in a refereed journal or conference proceeding.


Students are encouraged to choose research in areas closely linked to their professional experience. Unlike most students in traditional full time doctoral programs our students are formed in the profession and use their experience and knowledge as a platform for their research. Also, because of their senior positions in the work environment, they are able to manage and control studies and empirical research that would otherwise be difficult or impossible. This is an important strength of the DPS in computing.

The dissertation seminars in the third year are run very much like doctoral symposia at the major computing conferences, except that there are six of them throughout year 3 (the dissertation year). They provide fora for doctoral students to present their research and get detailed feedback and advice. All
the students and faculty advisors are present so that the advice spectrum is broad. The main objectives are:
• to allow students to practice early writing and effective presentation of their research progress;
• to provide a supportive setting for constructive feedback about students’ current research and guidance toward future research directions;
• to offer fresh perspectives on students’ work from faculty and students outside the research area;
• to promote the development of a supportive community of scholars and a spirit of collaborative research.

The student will give a presentation of 15-20 minutes followed by 15-20 minutes of questions and feedback. The experience is meant to mimic a “mini-” defense. Aside from the actual feedback, this helps the student gain familiarity with the style and mechanics of such a presentation. We use the Socratic method of clarification through confrontational dialogue. This forces the student to think independently and to vigorously defend a position.

Agile approach to the dissertation process
Since the agile methodology and the concept of being agile are important topics in the DPS program, it seemed only natural to approach the dissertation process in an agile manner. Typically the dissertation process is “waterfall- like” with heavy upfront planning and little writing. The main deliverable of the dissertation process is the dissertation manuscript. Before we adopted the agile approach, our students, too, looked at the dissertation as a very large task that was overwhelming.

What it means to be agile is defined in the Agile Manifesto for software development. [2] There is a set of important principles behind the agile manifesto. We have adopted some that we found to be the most important for our process:
• Early and continuous delivery of valuable product;
• Welcoming changed requirements, even late in development;
• Delivering working product frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale;
• Measuring progress primarily through useful deliverables;
• Using agile processes for sustainable development, enabling sponsors, developers, and users to maintain a constant pace indefinitely;
• Valuing simplicity as the art of maximizing the amount of work not done.

We had been using the agile approach in the later stages of the dissertation process with success. We are now starting to use it early in the process beginning with the idea paper and all the way through to the defense. We use some of the eXtreme Programming (XP) - like practices with descriptive terminology adapted for this non-software use:
• Communicating closely between advisor and student;
• Doing the simplest thing that will work with what you know now;
• Short time-boxed iterations of work;
• Continuous integration of accepted new work product;
• Pairing;
• Sustaining pace;
• Refactoring – restructuring and reorganizing.

We suggest a three-week iteration cycle. For each iteration there is a planning session during which the dissertation short-term deliverables (“stories”) to be done are discussed and estimated. The stories are written by the student and/or the advisor. Every story has a written deliverable. These may be annotations of readings or parts of the final dissertation manuscript. The stories are simple, focused and small.

Once the stories are estimated by the student, the student and advisor together select which stories to complete in the current iteration. Each iteration has a “velocity,” which is a measure of how much work that is completed and accepted can be done. The velocity of the next iteration is set to be the amount of work accepted in the prior iteration. This maintains a sustainable pace that the student is comfortable with and supports continuous delivery of quality product.

There is continuous communication between the student and the advisor; if the student determines that it will not be possible to complete all the stories for the iteration, then together they decide what to drop. Only completed and accepted work is integrated into the dissertation product. If the student finishes the stories before the end of the iteration, more stories are selected based upon their value and the velocity remaining in the iteration. Even though it might appear to be proscribed and overly structured, this process actually promotes creativity and helps many students finish in a shorter time period.

Success stories – student testimonials
Several students offer testimony to the impact of the DPS on their work practices and functionality:

Three students who work for a multinational computer technology and consulting corporation:

“I chose to look into Agile Software Development methods as my area of research... I’m clearly helping to shape how software is developed across the corporation by taking a leadership role in promoting, using and lecturing about agile methods.”

“Throughout the program, I was exposed to the latest software methodologies and technologies, and was able to capitalize on my own experience and the experience of other professionals; and I was able to witness and participate in innovative research... My own dissertation offered my employer the opportunity to witness groundbreaking research in a real world situation and reap the benefits of the experimentation when it proved successful.”

“Working within a major technology company, I was constantly encountering topics and situations that revolved
around new and emerging technology. ...I applied the learning from the DPS program immediately. It allowed me to more fully and more quickly understand the evolving technology I encountered and added tremendous value to that technology by more effectively applying my background in financial services.”

A student who is a senior vice president of a large insurance company:

“In the program we are surrounded by experts in many computing areas such as programming, engineering, teaching, management and others all united by common interests in conducting top-quality research in the world of computing. ... As a practical matter, I was able to use my research in Agile Development Methods to manage an enterprise-wide project, completing it on time and on budget. This success led to added responsibilities and more visibility by top executives at my firm.”

A student who is the CIO of a major food manufacturing company:

“*My Doctor of Professional Studies (DPS) in Computing will help me grow my company. How do you prepare for large-scale technological change on a national and global level? My DPS is helping.... I’m using skills I gained from my studies to drive the notions of “Fearless Change,” “Agile,” and “Retrospectives” with my IS/IT department. The Pace DPS program is one of the few doctoral programs for working professionals. It is the one that I needed.*”

**Completion and retention rate**

Every doctoral program will have students who have completed all but the dissertation (ABD); some say it is the nature of the beast. [7] What matters is how the program manages the ABD issue. It is ultimately the doctoral student’s responsibility to complete the dissertation. Even after the official matriculation for coursework has ended (in our case after the third year), it is still the responsibility of the program to make every effort to get a student through.

Why do some students drag on? The watch phrase in the DPS program is “life gets in the way”. We cannot help that, but we can provide a support group to help. We schedule sessions to bring the community back together to re-energize. If a student needs to take a leave due to work or personal issues, the community of support and the team structure of the work permit the student to defer study and rejoin a new class. This is why our retention rate is over 90%.

A summary of the program’s completion rate is given in Table 1 and Figure 1. Table 1 gives the completion rate for each of the classes – class of 2002 (the first class that started in 1999) through class of 2008 (with the class of 2009 completing as this is being written). This does not completely reflect the status of the dissertations. We never give up on a student. However we have lost a few, 7 to be exact who we believe will never go beyond ABD. But we have many more who are close to finishing and others who are making regular progress as shown in Table 1.

<table>
<thead>
<tr>
<th>Year</th>
<th>Number Completed</th>
<th>Number Expected to Complete within 6 months</th>
<th>Number Making Progress</th>
<th>Number Permanent ABD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>15 of 20</td>
<td>2 of 20</td>
<td>1 of 20</td>
<td>2 of 20</td>
</tr>
<tr>
<td>2003</td>
<td>6 of 17</td>
<td>4 of 17</td>
<td>6 of 17</td>
<td>1 of 17</td>
</tr>
<tr>
<td>2004</td>
<td>11 of 16</td>
<td>1 of 16</td>
<td>3 of 16</td>
<td>1 of 16</td>
</tr>
<tr>
<td>2005</td>
<td>11 of 18</td>
<td>0 of 18</td>
<td>5 of 18</td>
<td>2 of 18</td>
</tr>
<tr>
<td>2006</td>
<td>6 of 16</td>
<td>2 of 16</td>
<td>8 of 16</td>
<td>0 of 16</td>
</tr>
<tr>
<td>2007</td>
<td>8 of 16</td>
<td>2 of 16</td>
<td>5 of 16</td>
<td>1 of 16</td>
</tr>
<tr>
<td>2008</td>
<td>5 of 15</td>
<td>3 of 15</td>
<td>7 of 15</td>
<td>0 of 15</td>
</tr>
<tr>
<td>Totals</td>
<td>62 of 118</td>
<td>14 of 118</td>
<td>35 of 118</td>
<td>7 of 118</td>
</tr>
<tr>
<td>Total%</td>
<td>52%</td>
<td>12%</td>
<td>30%</td>
<td>6%</td>
</tr>
</tbody>
</table>

Figure 1 compares our completion rate against the traditional PhD rate recently compiled by the Council of Graduate Schools PhD Completion Project[5] Traditional doctoral programs often take students fresh from BS degrees, so it is understandable that the completion rate is spread out over a longer time interval than ours. The “DPS + 2 years” curve in Figure 1 shows how it compares with the traditional completions when shifted 2 years to consider the time from BS to MS. While we are doing better than the national average, we are always trying new methods to improve the completion rate and the dissertation quality.

**Conclusion**

Our approach to doctoral education is non-traditional mainly because we do not serve traditional doctoral students. Providing a superior doctoral experience for a diverse group of senior computing professionals for whom traditional programs are not possible or appropriate presents many challenges. While traditional doctoral study is a process of formation, the DPS in computing enhances and extends professionals who have already formed a professional identity, and empowers them to do greater things in their organizations and their future professional lives. The program exploits the depth and breadth
knowledge and experience that our students have in computing as a result of their professional experience, and takes a cross-cutting approach to curriculum, embracing such areas as agile software and non-software processes, networking and the Internet, design and organizational patterns, and security.

We believe that a student cohort is required. We learned early that more important than a cohort is a teaching-learning community. Students contribute their own knowledge and background experience to the learning of others. Faculty members act more as research and learning facilitators than instructors and traditional dissertation advisors. They are part of the teaching-learning community of support, and they actively interact with all the students, not just their advisees. All members of the community – students and faculty – are proactive in supporting and encouraging the dissertation progress of every student in the class. The synergy of diverse multidisciplinary yet deep individual knowledge and experience provides a support structure for timely completion.

We believe that we are creating a new class of scholarly professionals, who, acting as stewards of the computing discipline, are capable of critically evaluating new ideas and transforming and interacting with interdisciplinary professionals. They recognize the importance of representing and communicating ideas effectively, and thus effect immediate knowledge transfer to make significant changes in the way organizations function.

References

Fred Grossman (grossman@pace.edu) is a professor and Program Chair of the Doctor of Professional Studies in Computing at the Seidenberg School of Computer Science and Information Systems of Pace University in White Plains, NY

Charles Tappert (ctappert@pace.edu) is a professor of computer science at the Seidenberg School of Computer Science and Information Systems of Pace University in Pleasantville, NY

Joe Bergin (jbergin@pace.edu) is a professor of computer science at the Seidenberg School of Computer Science and Information Systems of Pace University in New York, NY

Susan M. Merritt (smerritt@pace.edu) is Dean Emerita and professor of computer science at the Seidenberg School of Computer Science and Information Systems of Pace University in Pleasantville, NY

To be published in the Communications of the ACM