Global J. of Engng. Educ., Vol.11, No.1
Published in Australia

Software Engineering Education for the Pervasive Human-Centric Computing Era

Gilda Pour
Department of Computer Engineering, San José State University
San José CA 95192-0087, United States of America

Moving away from decades of machine-centric computing and making pervasive human-centric computing, the new wave of computing, a reality revolutionises the relationship between humans and computing systems. In order to implement the vision of pervasive human-centric computing, it is necessary to reform software engineering education to well prepare graduates of software engineering programmes for the new challenges and opportunities of software engineering in the pervasive human-centric computing era. In this article, the author provides an overview of pervasive human-centric computing and the new challenges and opportunities associated with pervasive human-centric computing, and discusses the motivation for pervasive human-centric computing and the necessity of the reform of software engineering education to integrate pervasive human-centric computing into software engineering education and curricula. A wide range of technical and non-technical issues that need to be addressed by cross-disciplinary research groups, as well as a set of core strategies for the reform of software engineering education, are also presented.

INTRODUCTION AND MOTIVATION

The late Michael Dertouzos, Director of the MIT Laboratory (1974-2001) for Computer Science, who pioneered MIT Project Oxygen to make pervasive human-centric computing a reality, pointed out the need for pervasive human-centric computing. He stated the following:

> *If computers are to live up to the promise of serving us, they will have to change drastically and never again subject us to the frustrating experiences we all have shared* [1].

It is envisioned that pervasive human-centric computing systems will help people to achieve more while doing less. These systems will:

- Understand us when we speak to them;
- Do much of our routine brainwork for us;
- Get us the information we want, when and where we want it;
- Help us work with other people across space and time;
- Adapt on their own to our individual needs and desires [1].

In the pervasive human-centric computing era, we will not need to carry our own physical devices with us any more. Instead, configurable generic devices, either handheld or embedded in the environment, will bring computation to us, whenever we need it and wherever we might be.

As we interact with these *anonymous* devices, they will adopt our information personalities. They will respect our desires for privacy and security. We will not have to type click, or learn new computer jargon. Instead, we will communicate naturally, using speech and gestures that describe our intent (*send this to Hari* or *print that picture on the nearest colour printer*), and leave it to the computer to carry out our will [2].

Moving away from decades of machine-centric computing and making pervasive human-centric computing a reality to significantly improve the lives of people will indeed revolutionise the relationship that exists between humans and computers.

PERVASIVE HUMAN-CENTRIC COMPUTING

Pervasive human-centric computing systems will change how businesses, organisations and governments work with each other, as well as how individuals interact. It represents the dawn of a new era in Information Technology (IT) [1][3].

To shift the focus of computing from machines to humans, major changes are required not only in technologies and systems, but also in the approach to developing, deploying and managing technologies and systems. Mark Weiser, then Chief of Technology Officer (CTO) for Xerox's Palo Alto Research Center, stated the following:

> *The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it* [4].

Weiser presented his vision for pervasive human-centric computing in 1991. He further articulated his vision as follows:

> *There is more information available at our fingertips during a walk in the woods than in any computer system, yet people find a walk among trees relaxing and computers frustrating. Machines that fit the human environment instead of forcing humans to enter theirs will make using a computer as refreshing as taking a walk in the woods* [4][5].

Pervasive human-centric computing environments will no longer be virtual environments for storing and running software as do today's computing environments. Computing systems will no longer be machines that run programs in virtual environments as do today's computing systems. Software applications will no longer be written to exploit physical devices' capabilities as do today's software applications. Devices will no longer be reactive and managed by users as do today's devices [6][7].

Pervasive human-centric computing systems are proactive systems that are capable of sensing, measuring, monitoring, predicting and reacting to physical world conditions. To support a wide variety of human activities, pervasive human-centric computing systems must be:

- Pervasive: being available everywhere and accessing the same information base through every portal;
- Embedded: sensing and affecting the physical world;
- Nomadic: allowing users and computations to move around freely to meet the users' needs;
- Adaptable: providing flexibility and spontaneity in response to changes in the user's requirements and operating conditions;
- Powerful yet efficient: freeing itself from constraints imposed by bounded hardware resources, addressing system constraints imposed by user demands and available power or communication bandwidth;
- Intentional: enabling people to name services and software objects by intent;
- Eternal: never requiring shut down or reboot while components are added or removed in response to demands, errors or upgrades [2].

Existing applications for global networks, such as the Internet, must be modified to completely integrate pervasive human-centric computing devices into the existing social systems. Pervasive human-centric computing environments will have to deal with so many users, applications, various devices and intelligent spaces that interact on a scale far beyond what is experienced today. Thus, to implement the pervasive human-centric computing vision, the existing backbone IT infrastructure has to be extended to meet the anticipated demand, and many current technologies in the categories of device, network, software, user and perceptual technologies must be revamped and combined.

In a pervasive human-centric computing environment, device and network technologies will create a seemingly uniform computing space with heterogeneous and mobile computing systems, connecting dynamically changing configurations of self-identifying mobile and stationary devices to form collaborative areas. Pervasive human-centric computing should encompass every device with built-in active and passive intelligence. Networks will need to configure and reconfigure themselves automatically, as nodes appear, migrate or disappear.

Software technologies, such as agent technologies, will support adaptability of software systems to the needs and requirements of different users and environments, and also to the various changes and failures in the system with minimal user interventions and with no interruption of the services they provide. Agent components will act on behalf of users, devices and applications to effectively provide transparent interfaces between different entities in the environment, hence, enhancing the invisibility envisioned in pervasive human-centric computing [8-12].

In a pervasive human-centric computing environment, user and perceptual technologies will directly address human needs and consist of the following:

- Collaboration technologies: enabling the formation of spontaneous collaborative regions that accommodate the needs of mobile people and computations, and also provide support for recording and archiving speech and video fragments from a variety of sources and/or events;
- Knowledge access technologies: offering vastly improved access to information and customised to the needs of users (ie people, applications and software systems);
- Automation technologies: offering natural, easy-to-use, customisable and adaptive mechanisms for automating and tuning repetitive information and control tasks;
- Perceptual technologies like speech and vision technologies: enabling communication with devices, networks and software to extend the range of user technologies delivered to all places.

In order to advance the vision of pervasive human-centric computing towards technical and economic viability, it is required to adopt a multidisciplinary approach to address the issue of the proliferation of computational resources in the physical world, and to revamp and combine technologies such as sensing technologies, intelligent technologies, mobile technologies, wireless technologies, agent technologies, automation technologies, collaboration technologies, knowledge access technologies, speech technologies and vision technologies.

The key advances in the Internet, mobile and multi-agent technologies have paved the way that will lead to pervasive human-centric computing. However, some *major* technical and non-technical issues are yet to be addressed. These are elaborated on below.

Non-Technical Issues

Legal, ethical, social, cultural, political, educational, economic and organisational issues must be explored for creating pervasive human-centric computing systems and technologies.

Technical Research Issues

To address technical issues in pervasive human-centric computing, research directions, such as the following, have been identified and pursued:

- Development of platform-independent pervasive human-centric computing applications;
- Development of pervasive human-centric computing applications involving multiple components that are required to manage their configuration changes, for instance, by developing and using reflective middleware platforms;
- Coordination and integration of pervasive human-centric computing components in a way that address the issues of reliability, quality of service, invisibility and security in pervasive networking;
- Development of techniques and technologies to address the security and privacy issues in pervasive human-centric computing. Just as users must be confident of their computing environment's trustworthiness, the infrastructure must be confident of a user's identity and authorisation level before responding to the user's requests. It is difficult to establish this mutual trust in a manner that is minimally intrusive. At such a large scale, perfect trust among all parties is an impractical ideal. Trust boundaries then represent seams of discontinuity in the fabric of pervasive human-centric computing. Indeed, an inherent contradiction lies at the heart of the pervasive human-centric computing vision. On the one hand, a computing environment must be highly knowledgeable about a user to conform to his/her needs and desires without explicit interaction. On the other hand, a system that is truly pervasive will encompass numerous users, physical areas and service providers. The research issues include how to achieve a balance between conflicting criteria, such as proactive actions and privacy; security and cost-efficiency; and security, privacy and quality. Increasing awareness, maintaining an audit trail and creating a *sixth sense* are being explored to tackle the problem;
- Development of a robust agent-based system architecture that enables software to adapt to changes in location and the needs of a user to respond to component failures, as well as newly available resources, and also maintains the continuity of service even when the set of available resources changes as a result of a resource failure or a resource addition;
- Development of a middleware that provides transparent service to users by addressing the issue of heterogeneity and interoperability in pervasive human-centric computing environments. A major issue is to address the heterogeneity between intelligent spaces that offer different levels of infrastructural intelligence;

- Development of a middleware that interfaces between the networking and end-user applications running on pervasive devices, mediates interactions with the network on the user's behalf, and provides users the services and information offered by pervasive human-centric computing environments;
- Explicit distribution and installation of applications for each class and family across a wide geographic area as the number of devices grows, and the distribution and installation of applications become unmanageable;
- Development of systems and devices that perceive context to provide service in a pervasive human-centric computing environment. This is more complex than it is in mobile computing, which addresses location- and mobility-management issues in a reactive context mode (responding to discrete events). Many computing systems and devices today cannot sense their environments and, therefore, cannot make timely, context-sensitive decisions;
- Development of new techniques for a user interface to remind users that their locations are being monitored and also alert them when the trustworthiness of the entity performing that monitoring changes. Thus, more research on protocols and mechanisms for authenticating and certifying an individual's location at any given time is required;
- Development of a new paradigm to address the issues of the high degree of system complexity, rising costs of administering complex systems, lack of sufficient supply of trained system administrators, increasing size of software, numerous and disastrous failures and outages resulting from human errors, and major obstacles in handling the changes that affect the development, integration, deployment and management of complex systems. This has led to the emergence of autonomic computing.

Cross-Disciplinary Research and Development

To address technical issues in pervasive human-centric computing, cross-disciplinary research groups need to draw on a wide range of expertise in areas such as wireless and mobile computing, wearable computing, augmented reality computing, context-aware computing, location sensing, sensing, security and privacy, social sciences, psychology, linguistics, human computer interactions [9][13-31]. This includes the following important areas:

- Cross-disciplinary research among field experts in device, networking, software, user and perceptual technologies is needed to address the major issues of limits on size, weight, power and energy sources, low performance and reliability, and vulnerability to security and privacy violations in wireless, mobile, and handheld computing. Mobile agent technologies, as well as voice and data communication technologies, are needed to develop intelligent wireless, mobile and handheld devices to address the key issue of loss or degradation of wireless connections in wireless, mobile and handheld computing;
- Cross-disciplinary research is needed to address the issues including mobility, portability, usability, size, weight, safety, voice, vision, ruggedness, fashion and harsh operational environments, plus long battery life, in developing useful wearable computing systems;
- Cross-disciplinary research is needed as Augmented Reality (AR) technologies use virtual information representations (visual, aural or other) to enhance human perception, creating the illusion that the real world is visually merged with a virtual world; for example, combining AR with image recognition could pop up a *water me* reminder as a person looks at a thirsty plant;
- Cross-disciplinary research is needed as mobile AR requires both detailed geospatial databases that describe the fixed world and location-aware computing support to match the user's location with that description. For example, if fire-fighters could look at a burning building and see a representation of the activities on each floor, it might help in saving human lives. Mobile AR devices will be used to provide assistance to disabled people. The mobile AR devices that enhance human vision could be used by sight-impaired individuals to overcome many obstacles as they move around. In addition, high-resolution geospatial data could deliver key information about the immediate environment to mobile users through sounds or tactile feedback. Similar techniques could be employed to augment human hearing. Cross-disciplinary research in this area not only could make it possible for users to hear sounds outside their normal perceptual range or to mitigate hearing deficiencies, but also could provide added sensory input in situations where vision is being fully engaged;
- Cross-disciplinary research is needed to address various research issues, such as the following with regard to developing AR systems:
  - How to make a trade-off between conflicting criteria, such as AR's intense computational

demands and the need for small and light wearable computers;

- How to design mobile AR devices with the capability to determine those aspects of reality to augment with which information components, and help to identify the object of interest by providing users with the necessary information from several sources, including a comprehensive 3D model of the locale, dynamic information from sensors if the object pointed to is mobile, and information derived from an analysis of the image captured by the camera;
- How to design interfaces of mobile AR devices to help users specify what type of information they need.

- Cross-disciplinary research is required to address issues such as the design of user interfaces for mission-critical context-aware applications. Context-aware computing describes the special capability of an information infrastructure to recognise and react to real-world contexts. A context comprises any number of factors, such as user identity, current physical location, time of day, date or season, weather conditions and what the user is doing. The main goal of context-aware computing is to acquire and utilise information about the context of a device to provide services that are appropriate to particular people, place, time, events, etc. Context information is useful only when it can be usefully interpreted and it must also be treated with sensitivity;
- Cross-disciplinary research is required in location sensing to develop the following:

    - New user interface techniques in order to remind users that their locations are being monitored and alert them when the trustworthiness of the entity performing that monitoring changes;
    - System design techniques to provide end-to-end control of location information;
    - Efficient auditing mechanisms to record the exposure of location information;
    - Fine-grained access control mechanisms to allow the precise release of location information to only the right parties under the right circumstances;
    - Protocols and mechanisms to authenticate and certify an individual's location at any given time;
    - Techniques for rapidly calibrating an environment for specific location-sensing technologies to reduce costs.

- Cross-disciplinary research is needed to address issues in location-aware computing, which is a new class of computing based on mobile computing, context-aware computing, location sensing and wireless networking. This includes identifying how geospatial data are acquired, how and with what quality they can be delivered and how mobile and geographically distributed systems are designed. Location-aware computing systems respond to a user's location, either spontaneously (eg warning of a nearby danger) or when activated by a user request (eg is it going to rain in an hour?). Potential applications span virtually every aspect of everyday life, including a person's ability to respond appropriately to natural and human-induced hazards;
- Cross-disciplinary research is needed to address the issues in smart sensing, eg in smart dust (an immersive sensing technology), including legal or social obstacles to physically embedding dust sensors in a space, cost considerations for large spaces and environmental concerns regarding the safe disposal of smart dust. Smart dust, named for its small size and disposable nature, needs to be embedded in the space that is going to be sensed (eg to monitor the structural decay of bridges, highways and buildings over time due to different factors (eg corrosion of reinforcing metal) and detect overloads of design limits to prevent catastrophic damage;
- Cross-disciplinary research is required for remote sensing, aka non-immersive sensing. Brilliant rocks are developed using non-immersive sensing technology, and are being used in NASA's manned aircrafts, unmanned drones and aerial surveillance via satellite. Most astronomy involves non-immersive sensing-using techniques like spectroscopy. A combination of brilliant rocks and smart dust can be used as a nervous system for the physical world that has both always-on and on-demand components. This combination will offer a right balance of speed, accuracy, longevity and cost-effectiveness for some application domains;
- Cross-disciplinary research is required to address security and privacy issues raised in pervasive human-centric computing, and to achieve the necessary balance between security and privacy. A pervasive human-centric computing environment must acquire and maintain a substantial amount of personal information about a user to conform to the his/her needs and desires without interacting with the user before taking every step. This may violate personal privacy, and raise legal and ethical issues. With the current technologies, perfect trust among all parties is an unattainable ideal;

- Cross-disciplinary research is needed to advance autonomic computing, the emerging next generation of computing, which is in high demand for building a wide variety of mission critical systems in important application domains, such as healthcare, elderly care, defence, homeland security, space and planetary exploration, air traffic control, transportation, finance, e-commerce, e-business, manufacturing and many others that affect safety and security. Autonomic computing has been derived from the human body's autonomic nervous system, which governs the human body's involuntary vital functions, such as breathing, respiration, heart beat (rate) and body temperature. Inspired by the human body's self-regulatory nervous system, autonomic computing systems will be self-aware and able to self-manage (ie self-configure, self-optimise, self-heal and self-protect). It is envisioned that autonomic computing systems monitor the environment and resources to be managed, analyse the collected information, and use it to develop and execute plans to manage the system and the resources without requiring human intervention [25-31].

CORE STRATEGIES FOR SOFTWARE ENGINEERING EDUCATION REFORM

The suggested core strategies for software engineering education reform include the following:

- Redesign of software engineering curricula by integrating pervasive human-centric computing and autonomic computing into the curricula;
- Systematic integration of applied and experimental research in software engineering for pervasive human-centric computing into software engineering education;
- Industry-academic partnerships in both research and education;
- Engaging students in cross-disciplinary research and development;
- Institutional support and funding for cross-disciplinary collaborations in research and education;
- Fostering life-long learning;
- Systematic updating of the contents and structure of software engineering curricula.

It is necessary to restructure software engineering curricula by integrating pervasive human-centric computing and autonomic computing into the curricula [6][7][32].

The rapidly evolving and multidisciplinary nature of pervasive human-centric computing and autonomic computing requires the systematic integration of applied and experimental research into software engineering education to enhance students' learning experiences. Engaging software engineering students in applied and experimental research helps them to acquire invaluable experience that they cannot gain by simply reading technical articles and attending lectures.

It is also critically important to provide students with exceptional opportunities to gain experience in performing activities required for developing innovative engineering solutions and system prototypes. Such beneficial experiences also help software engineering students gain knowledge and an appreciation of the applied and experimental research required for writing research publications that offer insights into software prototype development and case studies.

To further enhance students' learning experiences, it is crucial to develop and nurture industry-university partnerships in research and education. This will also help students to work with industry sponsors while enhancing their hands-on experiences, as well as their technical competences and skills [6][7][32-36].

Furthermore, the multidisciplinary nature of pervasive human-centric computing requires collaborations in educational and research activities among field experts from different areas, as explained in earlier parts of this article. Hence, it is essential for engineering educational institutions to foster cross-disciplinary collaborations in research and education so that students can engage in collaborative, multidisciplinary projects with faculty and other field experts and professionals across various fields from universities, industry and research organisations. This will also help students learn and enhance their engineering knowledge and skills, as well as their professional skills (eg teamwork, written and verbal communications, etc).

Collaborative multidisciplinary projects require extra efforts to ensure effective and productive cooperation among all the people involved. Thus, it is critically important to change the culture, funding structure and faculty performance evaluation system in academia to provide the necessary institutional support and funding for cross-disciplinary collaborations among faculty from different departments, colleges, and universities, and other researchers from industry and non-academic institutions.

Due to the nature of software engineering for pervasive human-centric computing and autonomic computing, software engineers need to be strongly committed to life-long learning and regularly update their technical knowledge, competences and skills. To help graduates become self-motivated and life-long learners, it is critical to provide students with opportunities to acquire both the awareness of the necessity of life-long learning and the knowledge, skills and abilities to engage in life-long learning.

In order to ensure that software engineering educational programmes provide the best learning opportunities for students, it is crucial to maintain the flexibility of software engineering curricula, and to update systematically the contents and structure of the curricula.

CONCLUDING REMARKS

The vision for pervasive human-centric computing, the new wave of computing, cannot be implemented if software engineering education is not reformed to well prepare graduates of software engineering programmes for the pervasive human-centric computing era. In this article, the author discusses the necessity of integrating pervasive human-centric computing into software engineering curricula and presents a set of suggested core strategies for integrating pervasive human-centric computing into software engineering education.

The suggested core strategies include redesigning software engineering curricula to incorporate pervasive human-centric computing and autonomic computing into the curricula; systematically integrating pervasive human-centric computing and autonomic computing research into education; engaging students in applied and experimental research; establishing and nurturing industry-academic partnerships in research and education; providing institutional support and funding for cross-disciplinary collaborations in research and education; fostering life-long learning; and systematically updating the contents and structure of software engineering curricula to better prepare students for the new challenges and opportunities of software engineering in the pervasive human-centric computing era.

REFERENCES

1. Dertouzos, M.L., *The Unfinished Revolution: Human-Centered Computers and What They Can Do for Us*. New York: HarperCollins (2001).
2. MIT Project Oxygen, http://www.oxygen.lcs.mit.edu/
3. Satyanarayanan, M., Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8, **4**, 10-17 (2001).
4. Weiser, M., The computer for the twenty-first century. *Scientific American*, 265, **3**, 94-104 (1991).
5. Weiser, M., Some computer science issues in ubiquitous computing. *Comm. ACM*, 36, **7**, 75-84 (1993).
6. Pour, G., Pervasive computing reforming software engineering education. *World Trans. on Engng. and Technology Educ.*, 2, **3,** 357-360 (2003).
7. Pour, G., Restructuring software engineering education towards cross-disciplinary collaborations to create pervasive information systems and technologies. *World Trans. on Engng. and Technology Educ*., 3, **1**, 85-88 (2004).
8. Griss, M. and Pour, G., Accelerating development with agent components. *IEEE Computer*, 34, **5**, 37-43 (2001).
9. Pour, G. and Laad, N., Enhancing the horizons of mobile computing with mobile agent components. *Component-Based Software Engineering*, IEEE Computer Society (2006).
10. Pour, G., Internet-based multi-agent architecture for service delivery. *Internet Computing,* 1, **5** (2004).
11. Pour, G., Web-based multi-agent architecture for distributed collaborative software engineering. *Applied Computing,* 1, **3** (2004).
12. Pour, G., *Web-Based Multi-Agent Architecture for Software Development Formal Peer I nspection*. In: Mohammadian, M. (Ed.), Computational Intelligence for Modelling, Control and Auto-mation Series. Burke: IOS Press (2003).
13. Kindberg, T. and Fox, A., System software for ubiquitous computing. *IEEE Pervasive Computing*, 1, **1**, 70-81 (2002).
14. Want, R., Pering, T., Borriello, G. and Farkas, K., Disappearing hardware. *IEEE Pervasive Computing*, 1, **1**, 36-47 (2002).
15. Vaslu, L. and Mahmoud, Q.H., Mobile agents in wireless devices. *IEEE Computer*, 37, **2**, 104-106 (2004).
16. Satyanarayanan, M., Wearable computing: is it just hype? *IEEE Pervasive Computing*, 1, **4**, 2-3 (2002).
17. Starner, T.E., Wearable agents. *IEEE Pervasive Computing*, 2, **2**, 90-92 (2002).
18. Hazas, M., Scott, J. and Krumm, J., Location-aware computing comes of age. *IEEE Computer*, 37, **2**, 95-97 (2004).
19. Patterson, C.A., Muntz, R.R. and Pancake, C.M., Challenges in location-aware computing. *IEEE Pervasive Computing*, 2, **2**, 80-89 (2003).
20. Davies, N. and Gellersen, H.-W., Beyond prototypes: challenges in deploying ubiquitous systems. *IEEE Pervasive Computing*, 1, **1**, 26-35 (2002).

21. Patterson, C., Muntz, R. and Pancake, C., Challenges in location-aware computing. *IEEE Pervasive Computing*, 2, **2**, 80-89 (2003).
22. Satyanarayanan, M., Of smart dust and brilliant rocks. *IEEE Pervasive Computing*, 2, **4**, 2-3 (2003).
23. Satyanarayanan, M., Privacy: the Achilles heel of pervasive computing? *IEEE Pervasive Computing*, 2, **1**, 23 (2003).
24. Myles, G., Friday, A. and Davies, N., Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2, **1**, 56-64 (2003).
25. Pour, G., Expanding the Possibilities for Enterprise Computing: Multi-Agent Autonomic Architectures. *IEEE CS* (2006).
26. Pour, G., Exploring Multi-Agent Autonomic Architectures for Telehealth Systems. *Telehealth Series* (2006).
27. Pour, G., Multi-agent autonomic architectures for quality control systems. *Pervasive Computing* (2006).
28. Pour, G., Exploring multi-agent architectures for autonomic systems. *Pervasive Computing*, 2, **1** (2005).
29. Pour, G., Prospects for Expanding Telehealth: Multi-Agent Autonomic Architectures, *IEEE CS* (2006).
30. Kephart, J. and Chess, D., The vision of autonomic computing, *IEEE Computer*, 36, **1**, 41-50 (2003).
31. IBM Autonomic Computing White Paper. Automating Problem Determination: First Step toward Self-Healing Computing Systems (2003), http://www-03.ibm.com/autonomic/pdfs/Problem_Determination_WP_Final_100703.pdf
32. Pour, G., Expanding the horizons of software engineering education: integrating autonomic computing into the curriculum. *World Trans. on Engng. and Technology Educ.*, 5, **1**, 179-182 (2006).
33. Pour, G., Griss, M. and Lutz, M., The push to make software engineering respectable. *IEEE Computer*, 33, **5**, 35-43 (2001).
34. Pour, G., Engineering curriculum reform to introduce students to security and privacy in the Internet era. *World Trans. on Engng. and Technology Educ.*, 4, **2**, 285-288 (2005).
35. Pour, G., Agent-Oriented Software Engineering (AOSE): its emergence as a cornerstone of enterprise software engineering education. *World Trans. on Engng. and Technology Educ.*, 2, **2**, 225-228 (2003).
36. Pour, G., Component-based development refining the blueprint of software engineering education. *World Trans. on Engng. and Technology Educ.*, 2, **1,** 45-48 (2003).

BIOGRAPHY

Dr Gilda Pour is a software engineering professor at San José State University in San José, USA. She has received prestigious awards including the National Academies' National Research Council (NRC) Senior Research Associateship Award, NASA Research Associateship Award and NASA-Stanford Faculty Fellowship Awards. Dr Pour's research and industrial experience with Hewlett-Packard, NASA, Air Force Rome Laboratory, IBM and Sun Microsystems has been in the areas of software engineering for pervasive human-centric computing, autonomic computing, agent-oriented enterprise software engineering, and Internet and mobile computing.

Dr Pour has published extensively. She has made significant contributions to major international conferences and journals, as well as the high profile projects led by the IEEE Computer Society to establish the software engineering profession, accreditation criteria for software engineering programmes, and professional certification for software engineers. Dr Pour has also received leadership service awards. Her contributions to software engineering education have also been recognised. Dr Pour has also helped develop the software engineering programme at San José State University.