Grand Challenges Conference Application

Prof. Jack Stankovic
Department of Computer Science
University of Virginia
email: stankovic@cs.virginia.edu

Grand Challenge: How to cost-effectively build and maintain large-scale embedded computer systems that are highly dependable, secure, and maintain privacy?

Almost all of today's distributed system's technology assumes fixed (or infrequently changing) network topology, unlimited power, no mobility, ignores real-time constraints, and assumes a wired infrastructure. Also, they do not deal well with interfacing with sensors and actuators; rather they assume a workstation interface. In the embedded systems of the future, large collections of sensors, actuators, and cpus will communicate wirelessly, be mobile, have highly dynamic topologies, be required to meet real-time constraints, and require power management, new security techniques, and special emphasis on privacy concerns. Because of this large set of new requirements most of the previous distributed systems technology is no longer applicable. For example, consensus protocols developed in the past don't work in these environments, yet we need consensus and team formation properties. Another example, is that almost all networks use node IDs as a fundamental design strategy; but in these sensor networks we see that this is not useful and many people are working on geographic based routing instead. The combination of wireless communication and small capacity of each node in these embedded systems create especially difficult problems for many systems issues, especially, computer security. Also, real-time constraints and dependability concerns add difficulties that must be solved in highly error-prone environments and with limited node capacities. Basically, a large collection of very new system constraints require many new solutions.

Further, ten years out when molecular computers may come on the scene then there can be an incredible increase in computer power in extremely small devices. This will again shake up all the systems-level tradeoffs.

These smart sensor systems also pose many new challenges for the software engineering of them, for new concepts of "components" to make variations of the systems (as they are deployed in similar but different environments) cost-effective, for maintaining them, for monitoring them to determine if and how well they are working, etc.

In the future, if a person can walk into a smart hospital and have all the services happen reliably and with privacy and not realize there is a sophisticated embedded system behind the scenes, AND the system is general enough that the basic technology and software can be deployed in other similar smart environments (e.g., other hospitals and smart Universities), then this Grand Challenge will have been met.

Within the above context there is another grand challenge:

Grand Challenge: How do we control the aggregate behavior of large collections of sensors, actuators, and cpus?

When very large numbers of devices (1,000,000s) are involved we may have to consider notions of densities of devices rather than individual devices. We also must consider how devices can operate together to achieve the behavior we want. Research is needed into how to specify the behavior and then how to show how solutions being implemented actually can achieve that behavior. We will need to consider both transient and steady state behavior. I suggest that we need a theory and practice of feedback control applied to these dense networks. To address real-time constraints we may also need a new calculus of feasibility analysis where a designer can a priori determine if deadlines will be met by keeping the system within a multi-dimensional feasibility region. The key point is that we must be able to do this analysis without having to consider every detail. A basis for this work is seen in the notion of schedulability bounds where it is possible to say that a collection of periodic tasks will meet their deadlines as long as the utilization of the set is less than 69%. We do not need to know details about the tasks, etc., but just their aggregate property. In addition to new analysis techniques we must also consider how to implement resource management and control algorithms to achieve the aggregate behavior desired. Will as set of hierarchical feedback controllers work? Will they work in the presence of failure; traditional feedback control is not effective for large scale systems with many faults. Consensus algorithms can deal with failures, but are too costly. Perhaps a new theory and algorithms that combine feedback control and consensus can achieve the goals.

Final Note: These above systems can be part of many smart applications such as biological or chemical detection systems for homeland security, home health care for the elderly or disabled, learning environments, or even entertainment. They provide a new means to acquire and disseminate information. The implications--technically, ethically, and sociall--are enormous.

Short Bio:

Professor John A. Stankovic is the BP America Professor and Chair of the Computer Science Department at the University of Virginia. He is a Fellow of both the IEEE and the ACM. Before joining the University of Virginia, Professor Stankovic taught at the University of Massachusetts where he won an outstanding scholar award. He has also won the IEEE Real-Time Technical Committee's award for Outstanding Technical Accomplishments and Leadership in Real-Time Computing. He has also held visiting positions in the Computer Science Department at Carnegie-Mellon University, at INRIA in France, and Scuola Superiore S. Anna in Pisa, Italy. He was the Editor-in-Chief of the IEEE Transactions on Distributed and Parallel Systems and is currently the editor-in-chief for the Real-Time Systems Journal. His research interests are in deeply embedded smart sensor networks, real-time and embedded systems, operating systems, distributed computing, and ad hoc networks for pervasive computing. He has been involved with the construction of many research systems including the Spring kernel, SpringNet, the Spring scheduling VLSI chip, CARAT, RT-CARAT, BeeHive, etc. and while working in Industry worked on the Safeguard Anti-ballistic Missile System, implementing significant portions of the surveillance code that ran on the prototype system at Kwajalein in the Marshall Islands. Prof. Stankovic received his PhD from Brown University.