Software systems are continuously growing in size, complexity, distribution, heterogeneity, and numbers and types of users. Even more critical are the rapidly growing role and importance of software in human lives. We are surrounded by software-controlled devices and depend on them in virtually all of our daily activities. Airplanes, trains, automobiles, telephones (mobile and fixed), GPS receivers, cameras, televisions, home security systems, smoke alarms, air conditioners, and countless other conveniences of modern life are all either made possible or significantly improved—sometimes revolutionized—by software.

These software systems are different from typical desktop applications in that they are embedded in devices that are pervasive in our lives. Traditionally, such systems have been developed by engineers from other domains (e.g., electrical engineers), as a side effect of the physical device under production. However, as these systems have grown in size and complexity (e.g., it is not unusual for a TV set to run over a million SLOC) and have needed to interoperate ever more widely and seamlessly (e.g., a GPS receiver, new stereo system, or cellular telephone can all easily “plug into” a modern automobile), there is a growing understanding that software engineering principles and practices are important in this domain.

Development of large-scale software systems is typically approached from a software architectural perspective. Over the years, software engineers have developed many shared architectural concepts, patterns, styles, techniques, and tools. The objective of this special session is to explore existing software architectural approaches to pervasive systems, bring this area to the forefront, and motivate the software engineering research community to take a look at the many remaining challenges more intently.

Topics of interest include, but are not limited to
- Modeling and analysis of pervasive systems’ architectures
- Architectural styles and patterns for embedded and pervasive systems
- Architecture-driven implementation strategies for pervasive systems
- Application family- and domain specific-based approaches to pervasive systems
- Architecture-driven dynamic adaptation in pervasive systems
- Self-adaptive pervasive systems