

Statement of Research Interests

My general research interests focus on computer systems and networks, including distributed computing systems, pervasive and mobile computing, real-time and embedded systems, and wired and wireless networks. During my Ph.D. career in Computer Science, I have worked on designing and implementing self-adaptive computer systems that may diagnose, overcome, and adapt to adverse and rapidly changing computing environments.

A common assumption in traditional software development is that the change in computing environments is rare and slow. With tens of years of research in distributed computing system and mobile computing, and with today's developments in miniaturization, the computing environments seem to become more heterogeneous and dynamic. That is, the environments are composed of a wide range of devices, with diverse architectures and purposes, interconnected via wired or wireless networks. Under these environments, a priori knowledge of available resources is not always feasible during the development stage. As a consequence, some software, such as real-time systems, may fail to work correctly since the required resource is unavailable under some unexpected conditions, and development and deployment of such software become more difficult.

In my thesis research, I worked to remove a fundamental barrier limiting the feasibility of developing and deploying software under heterogeneous and dynamic environments: lack of a priori knowledge of resources that prevents developers from customizing and configuring the software for certain environments. I developed an approach in which the software is decomposed into components. Each component may have more than one implementations that are implemented to work correctly for certain environments. Software will probe the environment, choose the appropriate implementation for each component, be assembled from the chosen component implementations, and then begin to execute. To support this adaptive software architecture, I designed an adaptive software framework, ASAP, which includes

- APIs for assembly: That is, the APIs are used to probe environments, find the appropriate component implementation and assemble them into software.
- Component specification language: The markup language may be used for specifying how well a component implementation will perform under a given environment, and the relationships between components of hierarchical complex software architectures. The infrastructure of assembly will be automatically generated from the specifications.
- Self-adaption mechanisms on different environment change scale, i.e. how radically does the environment change.
- Automatic component distribution: If multiple target platforms are given, the components will be automatically distributed to these platforms and work collaboratively through remote method invocation mechanisms.

By taking advantage of these features, the computer systems I developed can diagnose the environment changes and change their execution behaviors without down-time, so it improves the flexibility and reliability of the systems.

Ren-Song Ko

ASAP has been tested successfully on several Linux systems, including PC, iPAQ, and Nomadic XR4000 robot. My other research interests grows out of my experience working on applying ASAP to these systems with different purposes and constraints. For example of pervasive and mobile computing, the target platforms usually have smaller form factors and limited resources. They also have more mobility, so the environment may change more frequently. Thus, the performance of the adaption may become a problem, since the adaption process requires intense computation and the results may not be finished before the next environment change happens. Fixing the performance problems that result from rapidly changing computing environments will require innovative work in several areas. These include techniques for optimization analysis of component specifications, better component implementation searching algorithm, and distributed assembly techniques. Furthermore, with mobility, the computer may be used for location dependent computing that brings research challenges in location sensing and components discovery. I am interested in developing techniques and tools in these areas that will be applicable to a wide range of computer systems.

Traditional real-time software development methodology is, with full knowledge of resource capability of target platforms, to minimize cost while maximize performance and compliance with real-time constraints through human effort and experimentation. Such a development methodology is time and cost consuming for heterogeneous and dynamic environments. ASAP has also been applied to real-time domain, in which a soft real-time application may adjust its output quality to meet its time constraints during the run-time without human intervention. I became interested in improving ASAP to hard real-time software.

The mission critical characteristics of hard real-time software exhibited performance predictability problems due to the lack of a priori knowledge of available resources. I am particularly interested in combining several techniques, such as assembly results caching, early constraint failure detection, and early adaption, to improve predictability of ASAP.

Heterogeneity does not only come from computer systems, but also network that involves different communication media, protocols and different types of data traffic requests from delay-tolerant data, such as text, to delay-sensitive data, such as real-time and multimedia. Making delay sensitive software self-adaptive will not be enough, especially on the open network where massive numbers of participants may join and leave at their will and the computing environment changes due to network traffic fluctuation may be more frequent than self-adaption can handle. This type of program will become widespread over the next several years since, with the advances in hardware technology, the demands increase for real-time capable software on mobiles devices and wireless networks. I am interested in the approaches that provide levels of QoS in networks, particularly wireless networks since it is more difficult to provide QoS compared to wired networks due to device mobility and high error rates in wireless networks.

The common thread running through the these areas of my research interest is the need for software to detect and respond to the rapid changes in computing environments. I have described specific issues that I currently find interesting, arising in computers that are getting diverse, smaller, and more mobility. The rate of technological change will

Ren-Song Ko

remain high so I expect to find other similarly interesting issues over time. One of the most important functions of academic research in systems is to anticipate the effects of coming technological trends, determining the best ways to respond in order to guide future industrial development efforts.

In general, I intend to work closely with hardware architects and software developers, as well as with specialists from other engineering disciplines. I have found this collaboration to be extremely productive during my graduate study at Michigan State University. It ensures that my work remains relevant to coming technological trends and changing application requirements, and creates the opportunity for me to affect both future system designs and software developments beyond the research community.