Software Engineering for Ambient Intelligence Systems

Thomas Fuhrmann
Universität Karlsruhe (TH)
76131 Karlsruhe
fuhrmann@ira.uka.de

Abstract: AmbiComp is a new research project that will invest about 30 person years into the development of a new and simple software engineering approach for mobile embedded interactive systems. In order to achieve its ambitious goals, it will combine research from different fields such as mobile peer-to-peer networks and operating systems. As a result, developing applications across multiple embedded devices shall be greatly facilitated.

Ambient intelligence is the vision that many embedded sensor and actuator devices jointly perform applications that enrich the quality of living and the efficiency of manual labour. In the recent years, especially the European Commission has funded several research projects along this vision. If, however, this vision shall become an everyday reality, we have to simplify the software engineering process for such massively distributed systems of low-resource embedded devices. AmbiComp is a new BmBF-funded research project that aims at providing a simple but powerful basis for writing programs that run on such ambient intelligence systems. Beginning in summer 2006, software engineering experts and distributed systems experts, both from academia and industry, will collaborate to design and create a distributed Java operating system together with an integrated software engineering process.

The core of the AmbiComp project is a novel self-organizing routing mechanism, scalable source routing (SSR), which efficiently provides the key-based routing (KBR) semantics for embedded devices with limited resources. With KBR, messages are routed towards hashed identifiers rather than hosts. As a consequence, KBR can address services and objects independent of their actual location. Unlike typical peer-to-peer KBR systems such as Chord or Kademlia, SSR is a genuine network layer routing protocol that self-organizingly provides ad-hoc networking between embedded devices. As has been demonstrated in extensive simulations [1,2] SSR can cope well with mobile devices and ungracefully leaving devices. Moreover, SSR can implicitly build aggregation and distribution trees.

The second important building block for AmbiComp is a resource-efficient Java virtual machine (JVM). Originally, it was developed for lab courses with inexperienced students [3]. With its help, students could load their programs onto a SD memory card and insert this card into the embedded device. Debug output and exceptions were shown on an integrated LC display. Compared to the state-of-the-art approach of flashing native programs into a microcontroller, this Java approach greatly simplified and sped up the development process. Furthermore, it turned out that this JVM already provided all basic operating system functionality as it is typically provided by a microkernel: (1) Memory protection is guaranteed by the JVM even in absence of a hardware memory management unit. (2) Processes and threads are supported natively by the JVM. Scheduling can be controlled at the bytecode operation level of the JVM independent of hardware support. (3) Inter-process communication is easy since virtual and physical memories are identical on the JVM level. As a result, this JVM provides a simple but powerful basis to run Java applications on simple microcontrollers.

The AmbiComp project shall combine SSR with this JVM: The JVM separates the memory that is visible to the Java programs from the actual physical memory level. Hence, object references can exploit the full identifier space of the KBR system and thereby seamlessly address remote objects without the need for any external translation mechanism. Moreover, due to SSR's inbuilt support for mobile nodes, this approach does not need any centralized components such as object brokers or service directories to support node mobility. Furthermore, SSR can automatically route requests to the nearest available service and thereby easily cope with ungracefully leaving nodes. Nevertheless, many research questions are still open. For example, it is still not fully clear how object locking shall be performed in that environment, and it needs to be studied how the effect of ungracefully leaving nodes can be confined to as few applications as possible.

In spite of these open questions, we believe that AmbiComp will help to simplify software engineering for interacting mobile embedded devices. The KBR semantics together with the JVM provide an effective basis to hide many aspects of the distributed system. We hope that - as a result - writing software for AmbiComp will become as easy as writing software for a PC. In order to further support the software development process, the AmbiComp project will also develop an Eclipse plug-in that is especially tailored to the needs of developers of embedded interactive applications. Together with application experts AmbiComp shall thus also produce various sample applications. Currently, we are looking for potential application experts from the field of mobile and embedded interactive systems.

Owing to the limited space of this extended abstract, we could not discuss related work. Clearly, many recent and not-so recent results provide important contributions to the problems address in the AmbiComp project. For example, a similar routing approach has been proposed recently by an independent group [4]. Other groups have pointed out the benefits of Java operating systems previously [5]. Furthermore, there has been a plenitude of distributed systems developed over the last decades. However, unlike AmbiComp, most of them rely on centralized components.

Conclusion: AmbiComp is a new research project beginning in summer 2006. It combines efficient key based routing at the network layer with a lightweight Java virtual machine. Thereby, the AmbiComp approach eliminates several problems that are typical to distributed systems with centralized components. As a result, we expect AmbiComp to provide a simple and powerful basis for the development of applications running on top of a distributed system of small, interacting embedded devices.

[1] Thomas Fuhrmann. Scalable Routing for Networked Sensors and Actuators. Proc. IEEE SECON 2005, Santa Clara, California, September 2005.

[2] Thomas Fuhrmann, Pengfei Di, Kendy Kutzner, and Curt Cramer. Pushing Chord into the Underlay: Scalable Routing for Hybrid MANETs Universität Karlsruhe (TH), Fakultät für Informatik, Technical Report 2006-12.

[3] Thomas Fuhrmann and Till Harbaum. A Platform for Lab Exercises in Sensor Networks. Technischer Bericht - 4. GI/ITG Fachgespräch Sensornetze Zürich, Schweiz, March 2005.

[4] Matthew Caesar, Miguel Castro, Edmund B. Nightingale, Greg O'Shea and Antony Rowstron. Virtual Ring Routing: Network Routing Inspired by DHTs, Proc. ACM SIGCOMM 2006.

[5] Michael Golm. The Structure of a Type-Safe Operating System. Doctoral Dissertation, 16 September 2002.