Distributed Embedded Systems with AmbiComp

Johannes Eickhold

jeick@so.in.tum.de
Self-Organizing Systems Group
Faculty of Informatics
TU Munich

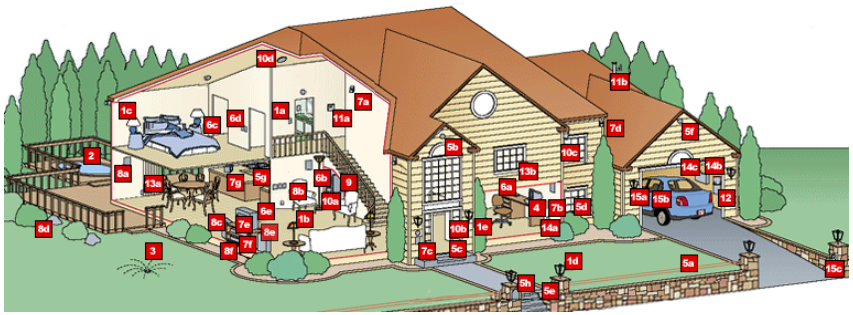
June 25, 2009

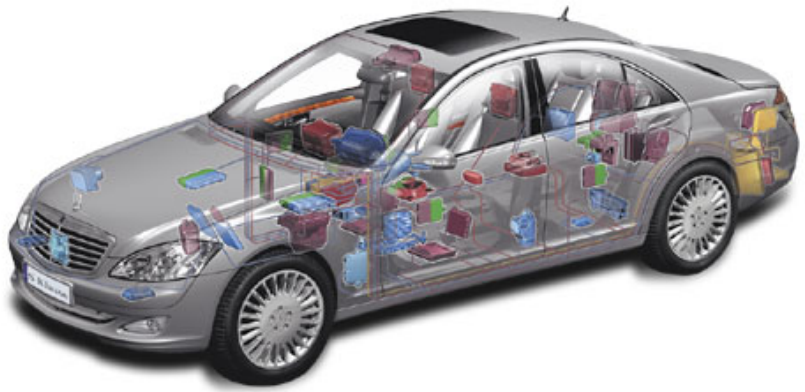
Motivation

Existing networks of embedded systems are designed for a specific purpose.

Motivation

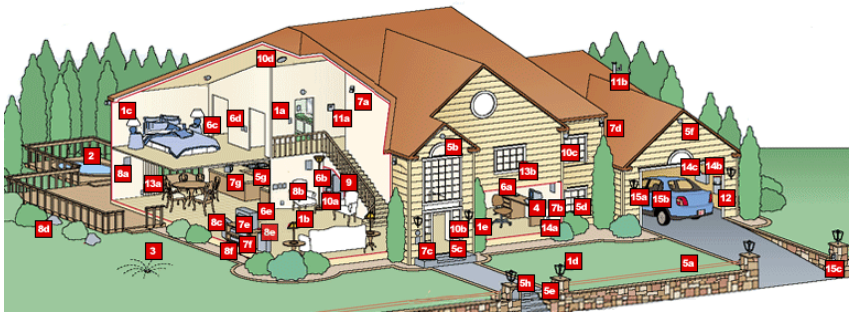
Existing networks of embedded systems are designed for a specific purpose.

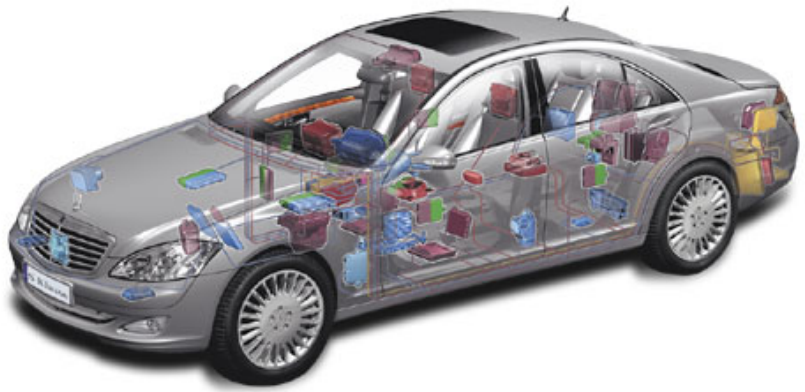




Motivation

Existing networks of embedded systems are designed for a specific purpose.





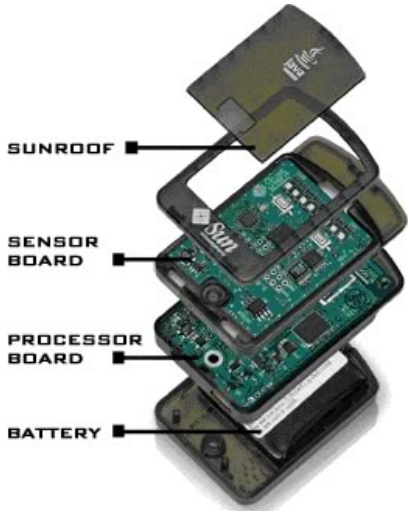
Which programming paradigms and infrastructure would be required by a ’general purpose’ network of embedded systems?

Overview

Related Projects

SunSPOTS

- 32-bit ARM processor
- Sun's SquawkVM (Java VM)
- SquawkVM written mostly in Java
- LED's, Temp. & Light Sensor
- Accelerometer and GPIO
- 802.15.4 radio



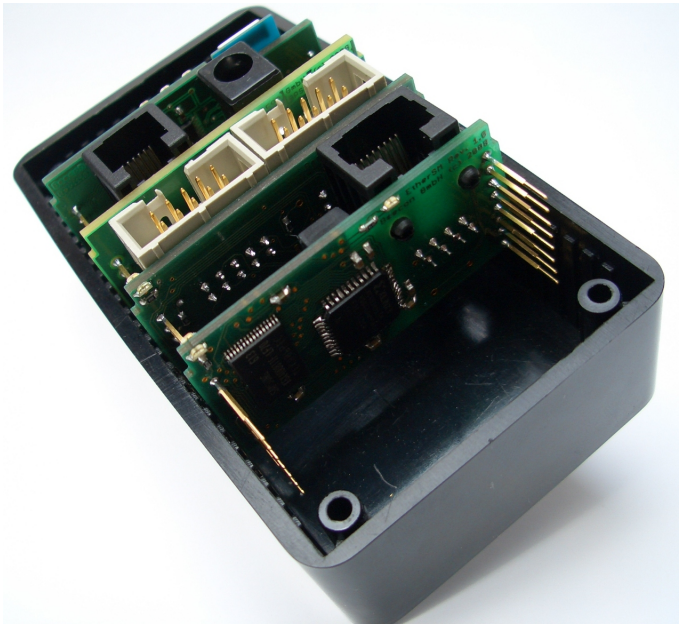
Sentilla Perk Kit (JCreate2)

- 16-bit micro-processor
- Java-based runtime environment
- LED's, two ports for Phidgets
- 802.15.4 radio

Ambient Intelligence Control Unit (AICU)

AmbiComp

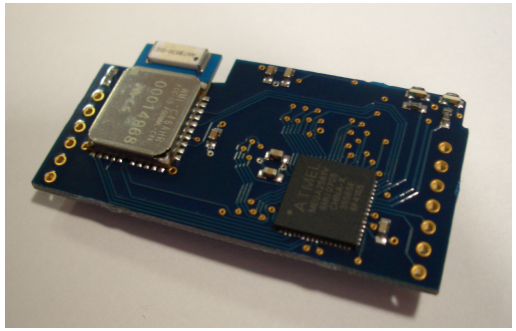
- 8-bit micro-controller
- AmbiComp VM
- Modular hardware
  sandwich modules (SMs)
- Stackable $\Rightarrow$ more functionality
- Well defined low-level interface:
  BIOS
- Seamless distributed operation
- Wireless and wired
  communication
- Power-over-Ethernet available
- Backplane for inter-SM comm.



Sandwich Modules

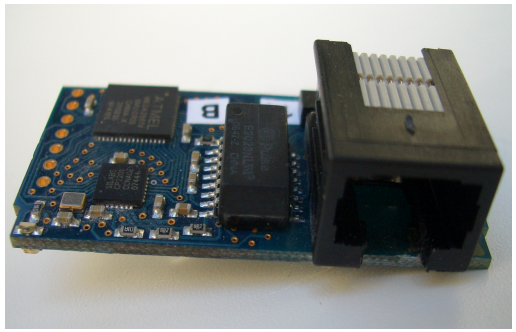
BTSM

- 8-bit AVR @ 7.37 MHz
- 256 KiB Flash, 512 KiB ext. RAM
- Bluetooth 2.0 radio



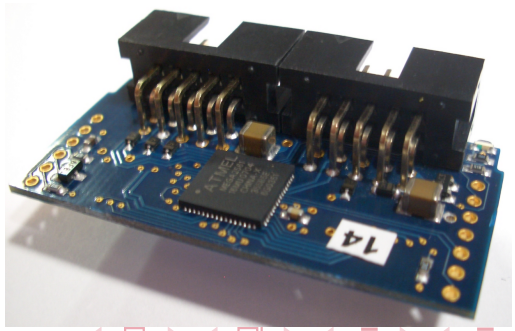
EtherSM

- 8-bit AVR @ 8 MHz
- 256 KiB Flash, 512 KiB ext. RAM
- Ethernet interface



IOSM

- 8-bit AVR @ 16 MHz
- 256 KiB Flash
- 16x dig. IO, 16x ADC, 4x DAC IO



Related Work

- Underlying concepts of **single system illusion** (SSI) well known from distributed cluster Java VM's: *cJVM (IBM), Jessica 2, Java Party, Kaffemik*
- Virtual Machine on bare metal - no OS required: *SquawkVM (Sun)*
- Distributed software modules: *Remote-OSGi and Concierge (ETH)*

AmbiComp aims at much **smaller** devices and much **larger** scale of distributedness

AmbiComp Virtual Machine

Characteristics of the AmbiComp distributed VM:

- Runs on 8-bit micro-controller
- Single system illusion
- Completely decentralized and self organizing
- Code-, object- and thread migration
- Runs across heterogeneous sandwich modules
- Eclipse as integrated development environment (IDE)
- Multi-threaded Java programming as if SMP
- External preprocessing tool: Transcoder
- Implicit communication: key-based routing

Self-Contained Sofware Stack

ACVM	executes transcoded Java programs transparently across the boundaries of SMs and AICUs
Comm. Stacks	contains communication protocol stacks such as TCP/IP and Bluetooth
BIOS	serves as hardware abstraction layer
Hardware	provides digital and analog IO, as well as communication interfaces

Single System Illusion (SSI) I

Questions in networks where no SSI is available:

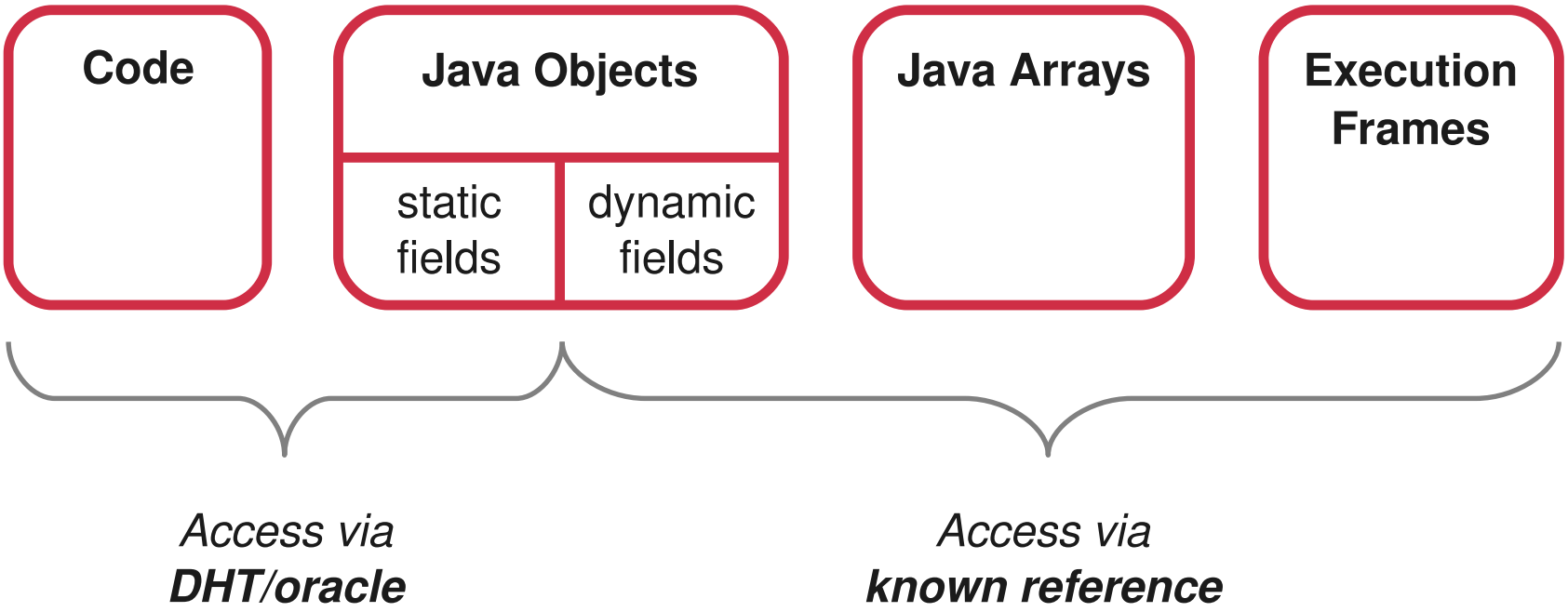
- Which resources are available in the network?
- Where are these resources located?
- What are the different components of the distributed application?
- How do these components communicate (RPC, RMI, proprietary protocol, . . .)?

Single System Illusion (SSI) II

The AmbiComp system:

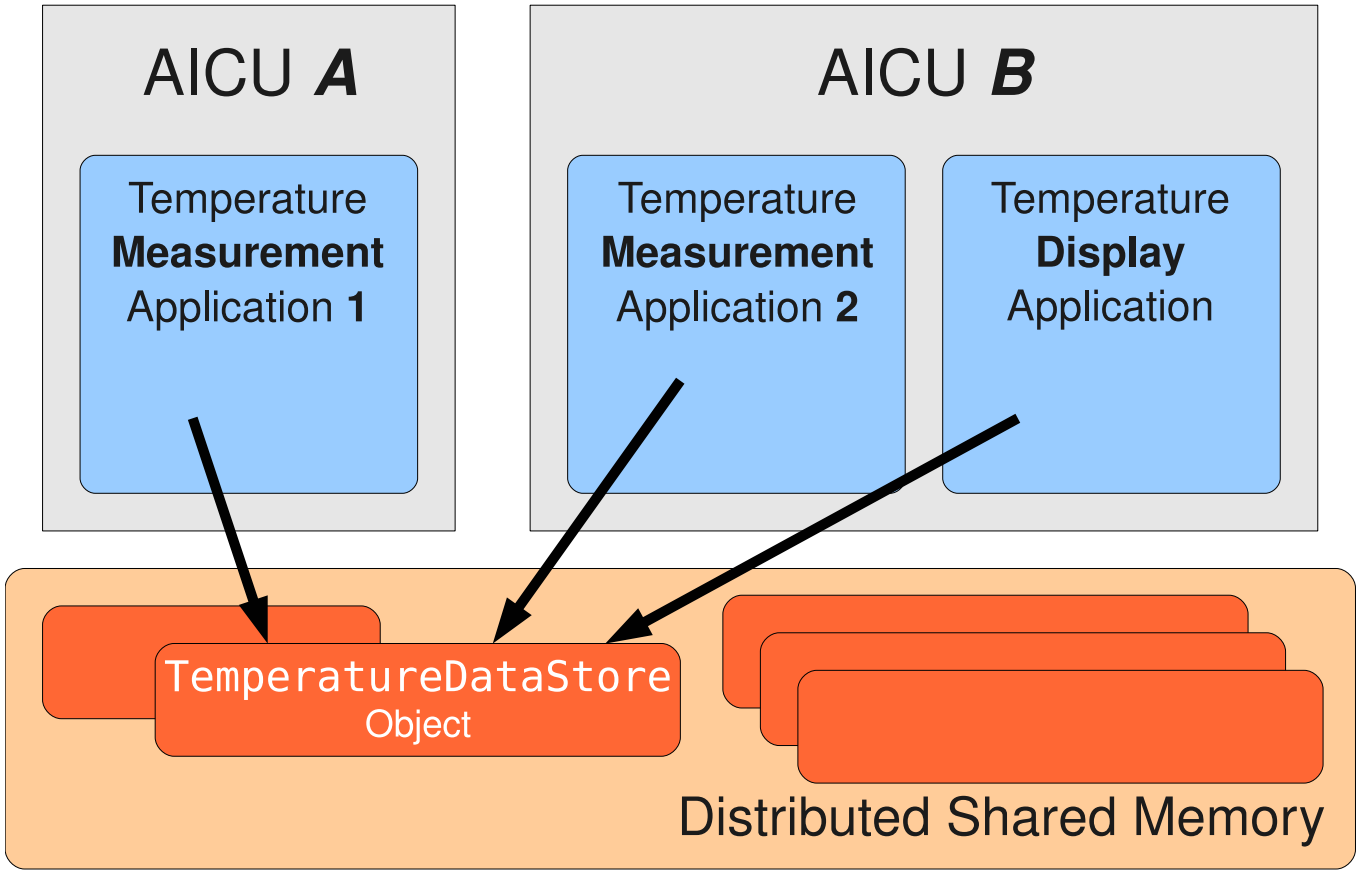
- hides the distributed and potentially heterogeneous nature of a networked embedded system
- shifts communication from explicit (i.e. by programmer) to implicit (feature of ACVM)
- adapts to changes in the network without need to adapt the software
- shares a global heap accross all ACVMs
- enables communication via GAOs in the global heap

Different Types of Global Accessible Objects (GAOs)



- Oracle implemented as distributed hash table (DHT)
- Reference passing via:
  - reading a field of an object or array
  - passing a parameter
  - using a method's return value

Application Example



Eclipse Plugin

Support developers where they need it most:

- Eclipse Plugin based on JDT
- Default set of APIs supports:
  Java platform (CLDC) and AmbiComp hardware
- Transcoding of Java applications $\Rightarrow$ BLOBs
- Deployment of BLOBs directly onto SMs
- Java in-system debugging via debug proxy and JDWP
- Transparent use of emulated and real AICUs
- Monitoring on SM level

Conclusion

AmbiComp:

- aims at easy development of distributed embedded applications
- creates modular hardware for rapid prototyping
- develops a compact distributed virtual machine
- offers single system illusion
- uses GAOs for implicit communication
- develops an Eclipse plugin to support application development

Thank you for your attention!

Questions?

References

http://www.ambicomp.org
http://www.beecon.de