Embedded System Development.
Java vs. C++

Dr.-Ing. Ingo Elsen, Dr. rer. nat. Hans-Jürgen Habrock, Dipl.-Ing. Detlev Uhlenbrock

T-Systems International Systems Integration

T-Systems
Embedded System Development.
Agenda.

- Software Development at T-Systems
- The Best Language for Embedded System Development?
  - Java for Embedded Systems – Pro/Contra
  - C++ for Embedded Systems – Pro/Contra
- Volere – Requirements and Constraints that drive the decision
- Example – Embedded Application with a large user base
  - Requirements
    - Basic function cores
    - Application Speed, Size etc.
  - Cost Drivers
  - Candidates to improve performance and reduce cost
- The decision, its impacts and its outcome
- Examples for Code migration
- Summary and Conclusion
T-Systems Systems Integration International Delivery Network.
We serve our customers all over the world.

- Concentration of industry know-how within P&D
- Industrialization of production within ADSF
## Industries

- Automotive
- Aerospace /Defense /Security
- Other Discrete Manufacture Industry
- Telecommunication
- Travel, Transport & Logistics
- Finance
- Public
- Healthcare
- Horizontals
T-Systems Systems Integration.
Industrialized Software Development.

- Project Management (PM Book)
- Software-Engineering (SE Book)
- Sales (Sales Book)
- ALM (SM Book)

- Requirement Analyses
- Specification
- System Design
- Programming
- Integration
- Rollout

- Configuration Management
- Quality Assurance
- Software Development Environment
- Reengineering
- Generators
- System Landscapes

T-Systems, Embedded System Development
Embedded System Development.
The Best Language for Embedded System Development.

- Runs Everywhere
- Real Object Oriented
- Easy to Learn
- You are Cool
- Complex Language
- Proved and Safe
- Rule the Hardware
- Old but Heavy
- If You Know the Language – You are Cool
- You get Everything
## Embedded System Development
### Java for Embedded Systems

<table>
<thead>
<tr>
<th>PRO</th>
<th>Contra</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Huge knowledge base</td>
<td>- Must have operation system (OS)</td>
</tr>
<tr>
<td>- Large number of frameworks and open source code available.</td>
<td>- Overhead for memory and microprocessor</td>
</tr>
<tr>
<td>- XML, Networking, Security and more are part of Java standard edition</td>
<td>- VM</td>
</tr>
<tr>
<td>- Write once, run everywhere</td>
<td>- Hardware abstraction layer</td>
</tr>
<tr>
<td>- Available Java optimized hardware (Java Microcontroller, J-Control, ...)</td>
<td>- Possible code optimization only with special compiler, no machine code</td>
</tr>
<tr>
<td>- Component based design with frameworks (e.g. OSGI etc.)</td>
<td>- No prediction of the runtime behavior except for Real-Time VMs based on RTSJ Specification</td>
</tr>
<tr>
<td>- Fast prototyping capability</td>
<td>- OS specific middleware for hardware integration needed</td>
</tr>
<tr>
<td>- Easy maintenance of Java based applications</td>
<td>- No direct access to hardware or OS functionality</td>
</tr>
<tr>
<td>- Large number of development tools also for testing, documentation and code-review</td>
<td>- Difficulty in source code protection (Reengineering)</td>
</tr>
<tr>
<td>- JME (Micro Edition) for embedded system development.</td>
<td>- ...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
## Embedded System Development

### C++ for Embedded Systems

<table>
<thead>
<tr>
<th>PRO</th>
<th>Contra</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Huge knowledge base</td>
<td>- Complex debugging (e.g. Pointer)</td>
</tr>
<tr>
<td>- Large number of frameworks and open source code available.</td>
<td>- Complexity of language can lead to code that is difficult to understand</td>
</tr>
<tr>
<td>- Generating of native machine code</td>
<td>- Not enough free or open tools</td>
</tr>
<tr>
<td>- C/C++ based operating systems</td>
<td>- Not platform independent</td>
</tr>
<tr>
<td>- No need for special hardware or runtime environments (VM etc.)</td>
<td>- Different compiler behavior (e.g. MS, Intel, gcc, Portland)</td>
</tr>
<tr>
<td>- Support for virtually every microcontroller</td>
<td></td>
</tr>
</tbody>
</table>
Java vs. C++

Example for an Embedded Development.
Embedded System Development.
Volere – Requirements and Constraints that drive the decision

Customer

Concept Phase

- Specification Phase
- Design Phase
- Build Phase
- Integration Phase
- Test Phase

- Business Case
- Feature List
- Wishes and Ideas
- Business Requirements
- 1. Level System Requirements
- 1. Level System Architecture
- Constraints

Target Costs
Embedded System Development
Basic Requirements and Constrains.

- Limited CPU performance
  - Power Consumption
  - Microcontroller vs. FPGA
- Small memory footprint (RAM, ROM, Flash etc.)
- Defined Hardware Interfaces
  - E.g. RS-232, USB, SPI, CAN, LCD, ADC/DAC, GPIO, SD-Card
- Small Size
  - Limited Space and Weight
- Limited Cost
  - Prototypes, Small number of Units, Mass-Products
- Software Update
  - Special Processes, Methods (e.g. SD-Card, USB)
  - Component Update or Complete Update
- Environment Constrains
  - Thermal, Size, Robustness, Usability
  - Size and Type of HMI (e.g. LCD, Keyboard)

Source: www.atmel.com
Example - Embedded Application with a Large User Base.
Introduction and Requirements.

**Example Road Charge System**

- Embedded system unit for road charging
  - Large number of units
  - Very reasonable price
- Essential Use Cases
  - Determine GPS position
  - Map geo-position to geo-object (e.g. roads)
  - Calculate the road charge
  - Communication via GPRS/GSM/UMTS with central system
  - Transfer road charge data to central system for billing
  - Update data for road charge models and maps
  - Software update
Embedded System Development.
Development Process.
Embedded System Development.
Development Process (Specification).

**Structure diagrams**
- Class diagram
- Component diagram
- Composite structure diagram
- Object diagram
- Deployment diagram

**Behavior diagrams**
- Activity diagram
- State diagram
- Use case diagram

**Interaction diagrams**
- Communication diagram
- Interaction overview diagram
- Sequence diagram
- Timing diagrams

**Requirements**
- Business Requirements
- Customer Requirements
- System Requirements
- Functional Requirements
- Performance Requirements
- Design Requirements
- Derived Requirements
- Allocated Requirements

**UML Models**
- Enterprise Architect

**Documentation**
- MS Office

**Specification**
- DOORS
Embedded System Development.
Dynamic Model View.

**Classic Process**
- Specification
- Implementation
- Code and binaries test and verification

**Model Driven Process**
- Specification
- Model test and verification
- Implementation
- Code and binaries test and verification

**Dynamic Model Test and Verification**

- **Use Cases**
- **OCL Expressions**
- **OCL Constraints**

- **UML Model**

- **Test & Verification**

- **Dynamic Model**
Embedded System Development.

- Specification
- Source Code
- Software Debugging
- Version and Configuration Management System
- Java IDE
  - Editor
  - Compiler
  - Debugger
  - Make, Build
  - Unit Test
  - Code Documentation
  - Code Review
- Byte Code
- Debugger Environment
  - Java VM

T-Systems
Embedded System Development.
Development Process – C++ (Implementation).

Specification

Source Code

C/C++ IDE

Software Simulator

Hardware Simulator

Software Debugging

Binaries

Hardware Remote Debugging

Version and Configuration Management System

C/C++ IDE
- Editor
- Compiler, Cross-Compiler
- Linker
- Debugger, Remote Debugger
- Make, Build
- Unit Test
- Code Documentation
- Code Review

C/C++ IDE

Software Simulator

Hardware Simulator
Embedded System Development.

Development Process – C++ (Software Remote Debugging).

Windows XP

Source

IDE C++
Eclipse

gdb – Debugger (Client)
Debugging Environment

Linux (Reference Platform)

Application
[C++]

API & Hardware Simulation

Ethernet
(TCP/IP)

gdb – Remote Debugger (Server)
Runtime Environment

Simulation Database

T-Systems
Embedded System Development.
Development Process – C++ (Daily Build Process).

- Preparation
  - Update source code and test data repository
  - Build binaries (Compile and link all sources)
- Performing test, review and verification processes
  - Verification of methods (Unit Test)
  - Verification of Unit Test coverage (LCov)
  - Low level use case and component test (Module Test)
  - Generation of source code documentation (Doxygen)
  - Code review (PCLint)
  - Source code metrics and quality analysis (RSM)
- Create reports
  - Evaluate, analyze report and results
- Performing source code modification and error correction
  - Code modification and documentation
  - Code error correction and documentation
Embedded System Development.
Cost Drivers - Candidates to Reduce Cost.

Java

C++

ARM9
400Mhz

CPU

ARM7
200Mhz

256MB

RAM

128MB

512MB

ROM

256MB

VM + JNI

VM

X

Framework

OSGI

X

T-Systems
Embedded System Development.
The decision, its impacts and its outcome.

**Decision**
- Software development with C++
- Development of the framework
- Use of Linux operating system
- Smaller memory footprint

**Impacts**
- Higher development costs
- Complex test and debugging
- Additional software simulations

**Outcome**
- Binaries with small memory footprint
- IP for framework and application
- Very reasonable price for the complete embedded system

**Contributions**
- Contribution = 50€/Unit
- Contribution = 25€/Unit
Embedded System Development.
Example: C++ Code Base – Code and Data Share.

**C++ Source Code and Data**

- Total 640 MByte
- Specification: 32
- Source Code Size: 12
- Unit Test Code Size: 160
- Module Test Code Size: 50
- Documentation: 320

**Embedded System Binaries**

- Total 157 MByte
- Operating System: 20
- Libraries: 8
- Driver: 13
- Framework: 4
- Application: 10
- API: 2
- Data: 100

- Without management data (PM/SE)
- Without miscellaneous code and data (e.g. simulation, scripts)
Embedded System Development.
Examples for Code migration.

Application
[C++]
Framework [C++]

API [C++]
Hardware Abstraction Layer

Drivers
e.g. Windows, Linux, VxWorks, QNX
Operating System

Embedded Hardware

GPS LCD Key GSM µWM IRM SD-Card SIM

Road Charge Application
- Pricing models
- User Interface (HMI)
- Road charge calculation

Services for Application
- Security
- Geo-mapping
- Diagnosis
- Navigation
- Communication
- Update

Application Program Interface
- Hardware and OS independent access to hardware modules, drivers and OS

Hardware Modules
- Customer specific modules
Embedded System Development.
Summary and Conclusion.

- **Java vs. C++**
  - *No winner* – chose the one that fulfills customers needs in the best way
    - One size doesn’t fit all
  - Do not chose your tools, architectures and designs by reflex
  - The system domain determines the best fit, that includes more than software, even more than hardware
  - Learn something about programming, how to choose the best fitting technology
  - Focused on technology issues but rarely considering things like maintainability and total costs

![Cost Share per Unit](chart.png)

(C++ = 100%)
Thank You...

Dr.-Ing. Ingo Elsen, Dr. rer. nat. Hans-Jürgen Habrock, Dipl.-Ing. Detlev Uhlenbrock

Ingo.elsen@t-systems.com

ADF ISS ES
Visit us at http://www.t-systems.com

Copyright © 2008 by T-Systems Enterprise Services GmbH. All rights reserved, also those regarding the reprint in extracts, the reproduction,copying (including microfiche and electronic reproduction) as well as the evaluation by data banks or similar systems.
Embedded System Development.
Services, IPC & Software Update – C++.

Update Process
- Stop service
- Kill service process
- Copy new service executable
- Execute new service
- Service connects to middleware
- Start service

Middleware
- Provides communication function for
  - Data exchange
  - Data object exchange
  - Message queue
- Provides status information about services
  - Start, running, stop

Application as services
- Execution of services as single applications
- Use of API function to access the operating system and hardware
- Use of middleware to communicate with other services

API
- Provides function to access the operating system and hardware

Drivers
e.g. Embedded Windows, Linux, VxWorks, QNX
Operating System
Embedded Hardware