Eclipse Tool Chain for Smart Development of High-Tech / Low Cost Embedded Systems
IS2T
Technologies
Tools
Design Concept
Benefits
References & awards
Nantes, 20 p., international presence

History
- 2004: birth of IST
- 2004-2006: Technology developments
- 2007-2008: MicroJvm on Blackfin, AVR, AVR32, ARM7, MIPS, ...
- 2009: Official launch at Embedded World and RTS

Venture Capital
ACE Management [Financière de B. / CEA / Thales / DCNS]

Management
Fred Rivard, PhD (Ex IBM, J9, Eclipse): CEO & CTO
Régis Latawiec (Ex ATMEL): Sales Director
François You: Financial Director

Partners
Technologies
IS2T Core Competencies

Object oriented technology provider for embedded systems

Object Oriented Languages

Compilers
Code Analysis
Optimization

Development Tools

Virtual Machines
Real-Time

Agile Oriented Processes
(Scrum – XP and Jazz)

IS2T owns 100% of its technology!
(IPs and know-how)
Java Platform = JVM + BSP

- **BSP gives partial hardware / OS abstraction**
  - Application binary program depends on hardware
  - Run-time depends on hardware and compilers

- **Platform gives total hardware / OS abstraction**
  - Independent programming environment (virtual processor)
  - Independent runtime environment (memory management, runtime errors, etc.)
OOP and VM for Embedded Systems!

- **Object Oriented Programming** for efficiency
  - Manage application architecture complexity
  - Increase engineering team productivity

- **Virtual Machines** for complexity abstraction
  - Rely on standardized services, not hardware specificities
  - Avoid software impacts when hardware changes
ESR – Embedded Specification Requests

• Why ESR?
  - Some JSR and Java technologies do not fit well to embedded systems:
    - High complexity and large memory footprint required
    - Focus on consumer applications such as cell-phones and PDA's.
  - IS2T provides open specifications for software libraries and technologies specifically targeted to embedded systems

• Examples
  - MicroUI™ (Micro User Interface)
    - A flexible graphical environment to design Human-Machine Interfaces
  - BON (Boot & Object Natures)
    - An improved data management with Immortal and Immutable objects
  - ECOM (Embedded Communication)
    - Support for UART, Ethernet, SPI, I2C and protocols such as TCP/IP
Industry Challenges

Cost & Risk management
Integration complexity
Long term solutions

Java benefits:
- Code quality
- Productivity
- Reliability
- Portability
- Scalability
- Maintainability
- Code compaction

Low-cost Micros
Small memory
Low performance
Real-time
C/asm legacy

Java benefits

MicroJvm benefits:
- High-speed
- Determinism
- Tiny footprint
- Interface to C / asm
- Low power
ECLIPSE-Based Tool Chain

JAVA
JDT
ICE TEA
IDT
C
CDT
IS2T
plugins
ANT
Toolkit
MicroEJ® for Java Applications

- **Java programming for embedded systems**
  - Write, debug and deploy within Eclipse IDE (=JDT project)

- **Optimize Java applications for MicroJvm®**
  - Byte-code optimization for MicroJvm® using SOAR®
  - Class-file tools for experts such as Classfile Inspector

- **Debug on simulator and targets** (Eclipse Launcher)
  - Debug at Java level on Smart Software Simulator (S3™) and targets
  - Perform static & runtime analysis on heap size,
  - Program coverage at binary level, etc.

- **Deploy on equipments**
  - In-Application Programming
• Simulate Java applications for embedded systems
  - Run Java applications on a simulated MicroJvm\textsuperscript{®} framework for workstations

• Hardware In the Loop simulation (HIL – mocks)
  - Simulate hardware for graphical Human-Machine Interface (LCD, touchscreen, keys, etc.)
  - Use physical communication channels such as USART, Ethernet
  - Interface to custom simulated peripherals (C, Java) over sockets

• Simulate embedded Java and native libraries
  - Java and IceTea\textsuperscript{©} native libraries run on S3\textsuperscript{™} to provide exact behaviour simulation
  - S3\textsuperscript{™} interprets IceTea\textsuperscript{©} routines as an extended Java language and provides common Java checks
MicroBSP® for Java Platform Design

- **Design your Java platforms**
  - Add native code implementations using Fast Native Interface (FNI™)
    - Reuse legacy C/asm libraries and design peripheral drivers (CDT)
    - Accelerate Java libraries

- **Access to IceTea® technology**
  - Write native implementations using Java-like language (IDT)
    - Object Oriented Programming (same Java syntaxe)
    - Structured Oriented Programming (struct and bitfield and interrupts)
    - Portable across platforms (no compiler semantic dependency)

- **Provide customized S3™ simulation environment**
  - Customize S3™ to simulate specific platforms including new peripherals, graphical displays, etc.
Example virtual Simulator
(design: 2 days)
Example real Simulator NXP LPC2478
(design : 2 days)
Java Platform Benchmarks

- **Design example**
  - Pixmap & vector icon 3D menu
  - Java only (no native specific)

- **Memory footprints (ARM7)**
  - (Java platform for this demo)
    - Code size: **177 KBytes**
      - (118 KB + 59 KB)
    - Ram size: **29 KBytes**
    - Includes vector drawings

- **Manpower**
  - 2 days for MicroUI™ training

---

### Code Size

<table>
<thead>
<tr>
<th></th>
<th>Code Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroJvm® + RTOS</td>
<td>57KB</td>
</tr>
<tr>
<td>Drivers + Native Libs</td>
<td>61KB</td>
</tr>
<tr>
<td><strong>Total Native</strong></td>
<td><strong>118KB</strong></td>
</tr>
<tr>
<td>Java Libs (CLDC+MicroUI)</td>
<td>59KB</td>
</tr>
<tr>
<td><strong>Total Java</strong></td>
<td><strong>59KB</strong></td>
</tr>
<tr>
<td>Total Constants</td>
<td>19KB</td>
</tr>
</tbody>
</table>

### Data

<table>
<thead>
<tr>
<th></th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native Heap + Stack</td>
<td>23KB</td>
</tr>
<tr>
<td>Thread Stacks</td>
<td>6KB</td>
</tr>
<tr>
<td><strong>Total Ram</strong></td>
<td><strong>29KB</strong></td>
</tr>
<tr>
<td>Fonts (+Arabic +Chinese)</td>
<td>19KB</td>
</tr>
<tr>
<td><strong>Total Constants</strong></td>
<td><strong>19KB</strong></td>
</tr>
</tbody>
</table>
Drag Emb'Drop

Design and test on PC

Simulated Java platform (PC)

Load binary image

Embedded Java platform (MCU)

Unified Java platform (PC + MCU)

Drag Emb’ Drop®

SimJPF

EmbJPF

Application

Middleware

HIL
OS
Mocking
ESR
BON
MicroUI
ECOM
JSR
CLDC
MIDP
IMP
OSGi

Micro
Jvm

Drivers

FNI
RTOS
MicroEJ allows the parallelization of software and hardware design phases, for a shorter global design process lead time.
• MicroBSP allows the customization of Java Platforms for different targets, in order to reuse a previous application and launch more products during the same period of time.
Our 4 Impacts on Value Chain

1. Reduced Investment
2. Increased Sales
3. Increased Margins
4. Reduced Risks
REFERENCES & AWARDS
- Defense / Space / Avionics
- Automotive Electronics
- Home Automation
- Smart Metering / M2M
- Telecom
- Medical
- Industrial Control
- Security
Trophy
“Best Technology for Embedded Development”
9 June 2009
www.is2t.com
http://edu.is2t.com