

Java In The Small: Enabling Standard Java On Embedded Devices Through Customization

Alexandre Courbot (LIFL)

Gilles Grimaud (LIFL)

Jean-Jacques Vandewalle (Gemplus Research Labs)

December 7th, 2004



Java does not fit into embedded devices

Embedded devices are hungry for Java, but...

- A complete JRE is too large for an embedded device.
- Stripped-down versions of Java have emerged:
 - Java Micro Edition,
 - Java Card,
 - ...



Problem

These versions are degraded right from the specification.

Our Proposal

Tailor a full-fledged Java Runtime Environment during its deployment according its runtime usage.

Java does not fit into embedded devices

Embedded devices are hungry for Java, but...

- A complete JRE is too large for an embedded device.
- Stripped-down versions of Java have emerged:
 - Java Micro Edition,
 - Java Card,
 - ...



Problem

These versions are degraded right from the specification.

Our Proposal

Tailor a full-fledged Java Runtime Environment during its deployment according its runtime usage.

Java does not fit into embedded devices

Embedded devices are hungry for Java, but...

- A complete JRE is too large for an embedded device.
- Stripped-down versions of Java have emerged:
 - Java Micro Edition,
 - Java Card,
 - ...



Problem

These versions are degraded right from the specification.

Our Proposal

Tailor a full-fledged Java Runtime Environment during its deployment according its runtime usage.

Java does not fit into embedded devices

Embedded devices are hungry for Java, but...

- A complete JRE is too large for an embedded device.
- Stripped-down versions of Java have emerged:
 - Java Micro Edition,
 - Java Card,
 - ...



Problem

These versions are degraded right from the specification.

Our Proposal

Tailor a full-fledged Java Runtime Environment during its deployment according its runtime usage.

Java does not fit into embedded devices

Embedded devices are hungry for Java, but...

- A complete JRE is too large for an embedded device.
- Stripped-down versions of Java have emerged:
 - Java Micro Edition,
 - Java Card,
 - ...



Problem

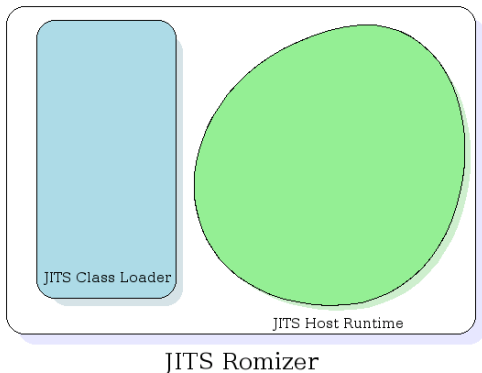
These versions are degraded right from the specification.

Our Proposal

Tailor a full-fledged Java Runtime Environment during its deployment according its runtime usage.

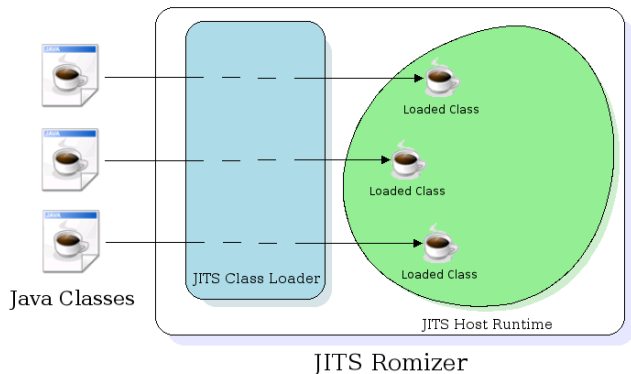
Romization

The JITS romizer generates a ROM image of an embedded system from a set of Java classes.



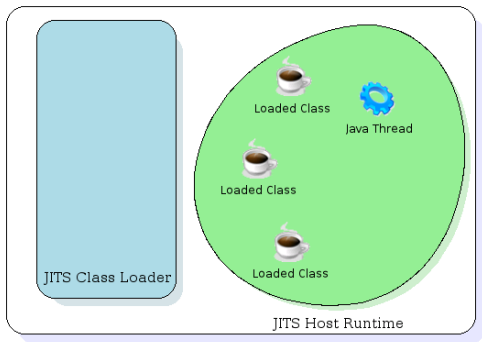
Romization

The JITS romizer generates a ROM image of an embedded system from a set of Java classes.



Romization

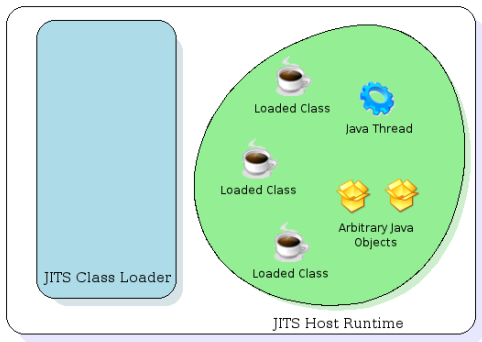
The JITS romizer generates a ROM image of an embedded system from a set of Java classes.



JITS Romizer

Romization

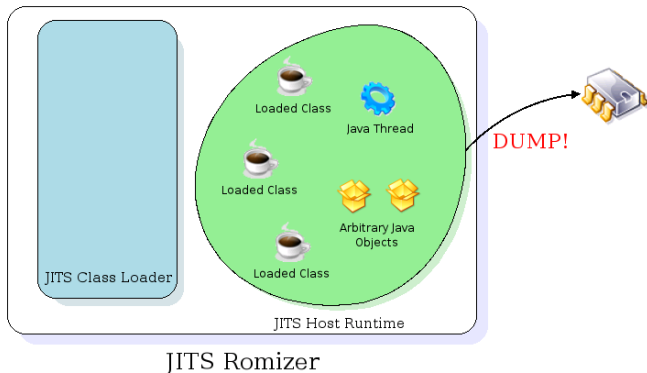
The JITS romizer generates a ROM image of an embedded system from a set of Java classes.



JITS Romizer

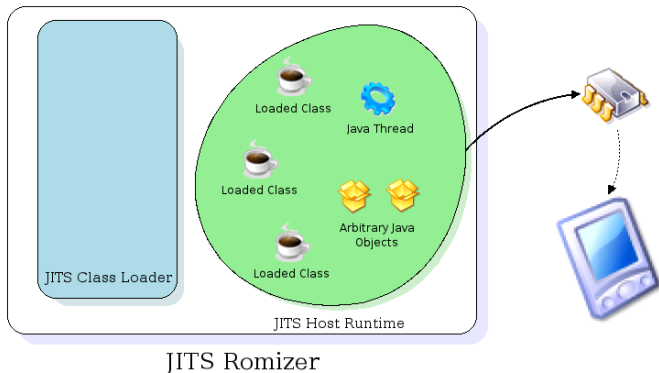
Romization

The JITS romizer generates a ROM image of an embedded system from a set of Java classes.



Romization

The JITS romizer generates a ROM image of an embedded system from a set of Java classes.



What do to during romization

- Romization brings great conditions for customization:
 - Complete view of the system,
 - Knowledge of its entry points (threads),
 - Ability to manipulate the system (host introspection).
- What can be done:
 - Constant pool packing
 - System components configuration
 - Abstract interpretation
 - Program proof

Our goals

Demonstrate that customizing during romization eliminates most of Java overhead and gives results comparable to stripped-down versions of Java.

What do to during romization

- Romization brings great conditions for customization:
 - Complete view of the system,
 - Knowledge of its entry points (threads),
 - Ability to manipulate the system (host introspection).
- What can be done:
 - Constant pool packing System components configuration
 - Abstract interpretation Program proof

Our goals

Demonstrate that customizing during romization eliminates most of Java overhead and gives results comparable to stripped-down versions of Java.

What do to during romization

- Romization brings great conditions for customization:
 - Complete view of the system,
 - Knowledge of its entry points (threads),
 - Ability to manipulate the system (host introspection).
- What can be done:
 - Constant pool packing **System components configuration**
 - Abstract interpretation Program proof

Our goal:

Demonstrate that customizing during romization eliminates most of Java overhead and gives results comparable to stripped-down versions of Java.

What do to during romization

- Romization brings great conditions for customization:
 - Complete view of the system,
 - Knowledge of its entry points (threads),
 - Ability to manipulate the system (host introspection).
- What can be done:
 - Constant pool packing System components configuration
 - Abstract interpretation** Program proof

Our goal:

Demonstrate that customizing during romization eliminates most of Java overhead and gives results comparable to stripped-down versions of Java.

What do to during romization

- Romization brings great conditions for customization:
 - Complete view of the system,
 - Knowledge of its entry points (threads),
 - Ability to manipulate the system (host introspection).
- What can be done:
 - Constant pool packing System components configuration
 - Abstract interpretation **Program proof**

Our goal:

Demonstrate that customizing during romization eliminates most of Java overhead and gives results comparable to stripped-down versions of Java.

What do to during romization

- Romization brings great conditions for customization:
 - Complete view of the system,
 - Knowledge of its entry points (threads),
 - Ability to manipulate the system (host introspection).
- What can be done:
 - Constant pool packing System components configuration
 - Abstract interpretation Program proof

Our goal:

Demonstrate that customizing during romization eliminates most of Java overhead and gives results comparable to stripped-down versions of Java.

Thank you for your attention...

Contact: Alexandre.Courbot@lifl.fr
 <http://www.lifl.fr/~courbot>

JITS Home Page: <http://www.lifl.fr/RD2P/JITS>