Project Background

- European Commission funded under the IST program
- 30 month project starting in January 2002
- Involving 4 companies and 2 universities from 4 European Countries
- Coordinator: FZI-Universität Karlsruhe, Germany
- Aicas, Germany
- Rose Informatik, Germany
- Linköpings University, Sweden
- Aonix, France
- Skysoft Portugal, Portugal

- My company is a recent PME (started 1998) that specializes in R&D and software development for aerospace.
Motivation

- Embedded software is an expanding field.
- It is expected that the next generations of embedded software will be integrated in distributed systems.
- Embedded software development requires qualified programmers.
- The need for embedded software is expected to grow at a faster rate than the number of qualified programmers.
- A “safe” language could assist the development process by requiring less qualifications from the programmers.
- Java could be such a language, but it was not designed with real-time requirements.

Solution

- Remove non-determinism from Java
- Provide a development environment that can assure the developer that the system is compliant with its performance requirements.
- Provide a communications infrastructure that is both simple to use and can provide real-time guarantees, being also fault-tolerant.
Goals

Java

- Integrated Tools
- Deterministic Implementation
- Distributed Real-Time Event Handling
- Real-Time Modeling and Analysis
- Adaptation to Limited Resources

Type
- Safety
  - Safe, Automatic Memory Management

Standard Libraries
- Integrated Tools
- Deterministic Implementation
- Distributed Real-Time Event Handling
- Real-Time Modeling and Analysis
- Adaptation to Limited Resources

Multi-processing
- Platform Independence

Object-Oriented Methodology
- Standards

HIDOORS - A High Integrity Distributed Deterministic Java Environment
João Ventura Skysoft Portugal
15:30 7-1-2002, Slide 5

Development Blueprints

- Optimizations
  - Static Garbage Collection
  - Garbage Collector Support
  - Native Backend
  - Java-to-C Compiler

- RT Date Access Standard
- Standard Classes Subset

- Real-Time Standard Classes
- Real-Time Network Classes

- Distributed Event Handler
- HIDOORS Real-Time Java Development Environment
  - Optimizing Compiler for Deterministic Code Generation
  - Distributed Event Handler
  - Standardized Real-Time Library Classes
  - Java Virtual Machine
  - RT GC

- Real-Time Data Access Standard
- Standard Classes Subset

- Real-Time Model Checker
  - Real-Time UML
  - URL Tools

- Integrated Development Environment
- HIDOORS Enhancements
  - Existing Code Base

- Garbage Collector Enhancements
- Real-Time Garbage Collector

- WCET Analysis Tools
- Profiling & Analysis Tools

- IDE extension

João Ventura Skysoft Portugal
15:30 7-1-2002, Slide 6

HIDOORS - A High Integrity Distributed Deterministic Java Environment
**Integrated Development Environment**

- Graphical UML Tool enhancing current tool from Aonix
- The sub-tool that allows specification of real-time automata will be enhanced with a component that will enable the checking of real-time temporal logic specifications, based on previous work performed at Univ. Karlsruhe.
- Java code Generator will be added to Aonix’s tool, with support for real-time automata and an analysis toolbox.

---

**Java Compiler**

- An optimizing compiler will be developed during HIDOORS with the objective of minimizing both the footprint of the Java application and the use of the garbage collector.
- The compiler will be enhanced with support for deterministic garbage collection. This is based on synchronization points for constant-time exact root-scanning of stacks and registers, write-barriers and a non-fragmenting object model.
- The existing base compiler uses C as a portable intermediate language. During HIDOORS, this will be enhanced to support native code generation.
Jamaica VM

- Java-Implementation based on JDK 1.2
- Exact real-time garbage collector without GC interruptions and guaranteed allocation time of a few µs.
- Builder Tool to create small and efficient applications for embedded systems.
- Java Native Interface to access existing code

runtime Environment

- Improve the Jamaica VM to support deterministic execution (deterministic constant time execution of all Java primitive operations)
- Improve the Garbage Collector, so that garbage collection overhead is minimized. Worst-case execution times for garbage collection activities will be determined to serve as input to WCET analysis tool.
- Enhance the Java monitors as validated by state-of-the-art research results.
- Modify subset of Java standard libraries to provide deterministic execution time for all operations that will be used in real-time code.
- Provide support for direct hardware access.
Timing Analysis

- In order to efficiently model the underlying architecture, a system model including the different structures of system caches and processor architectures will be developed. This has to be sufficiently generic to allow reuse of the model in different systems.

- Develop a WCET analysis tool based on the technology developed at the Univ. des Saarlandes, making use of the strong typing properties of Java.

Real-Time Network Communications

- Distributed event control will provide a manager that handles events with strict timing guarantees. This manager will be based on the CORBA event manager, but as a distributed service.

- Support for deterministic network protocols via a Java TCP/IP-like socket interface such as CAN or AFDX.

- Modify RMI functionality so that emphasis is placed on guaranteeing performance as required by real-time applications.
Future Work

• Official HIDOORS kick-off in the 15th of January
• Development is to be completed in the next 2 ½ years.

• Further results will be presented at dedicated workshops at related conferences or trade shows.

• Experiences will be fed to the J-Consortium standardization process for real-time Java.

• Public results will be disseminated through the HIDOORS website: http://www.hidoors.org