Preliminary Feasibility Analysis of Component Based Modelling and Automatic Java Code Generation for Nanosatellite On-Board Software

Óscar Rodríguez Polo
SRG-UAH
(Space Research Group – University of Alcalá)
Overview

- Motivation
- Nanosat1B on board sw component model
- EDROOM
- Feasibility analysis of EDROOM to JEDROOM
- Achievements and future work
Motivation

- Nanosat1B (INTA) on-board sw was developed:
  - Graphical Component Modelling
  - Automatic C++ Code Generation
  - EDROOM CASE tool assists in this process
- Nanosat 2 (INTA) will be based on LEON arch
- Currently there is Java support for LEON (FijiVM)
- ¿Could we port EDROOM for Automatic Java Code Generation on LEON arch?
- ¿Is feasible the modelling and automatic java code generation of a Nanosatellite on-board sw?
Nanosat1B

Weight ≈ 25 Kg

RAM: 1 Mb
PROM: 256 Kb
EEPROM: 256 Kb
MMU: 4 Mbyte

CPU: μc 68332
Clk: 16 MHz

Launched 31/07/09
43,000 sw code lines
Nanosat1B Sw Component Model
Nanosat1B Sw Component Model

STATE MACHINE DIAGRAM
EDROOM

1 Graphical definition of Structure and Communication

2 Behaviour Specification

3 Detail level implementation

4 Automatic Code Generation

Services Request Ext Sw Modules
EDROOM

AUTOMATIC CODE GENERATOR

EDROOM SERVICE LIBRARY

BASIC PRIMITIVES

REAL TIME

Components Generated Code
(platform independent components)

EdROOM Service Library

Top Layer
(platform independent)

Pr_Task, Pr_Semaphore, Pr_Time,
Pr_IRQ_Event, Pr_IRQ_Manager

RTOS
Feasibility Analysis of JEDROOM

JAVA CODE GENERATION

JAVA EDROOM SERVICE LIBRARY

BASIC PRIMITIVES

REAL TIME JAVA

JAVA VM (FIJI VM)
JEDROOM Requirements

- Memory management:
  - ESL receives the memory for the main function, so it can be compatible with scoped memory, immortal memory or heap memory provided by Fiji VM.

- Mutual exclusion access:
  - ESL requires the definition of critical section free of priority inversion. Fiji VM can be configured with --lock-impl pip flag for providing priority inheritance to synchronized methods.
JEDROOM Requirements

- Dynamic thread priority configuration:
  - ESL component threads (Pr_Task) require their priority can be assigned to the most prioritized message received.
  - The Fiji VM package `com.fiji.fivm.ThreadPriority` provides the `setPriority` method of `Thread` class that can be applied in order to get this functionality.

- Counting semaphores:
  - ESL requires counting semaphores (Pr_semaphore) that can be implemented in Java using the standard primitives `notify` and `wait`
JEDROOM Requirements

- **Timing:**
  - ESL (Pr_Time) requires both absolute and relative delays.
  - The Fiji VM package `com.fiji.fivm.Time` provides the method `Thread.sleep` for relative delays, while `com.fiji.fivm.r1.fivmRuntime` provides `fivmRuntime.waitAbsolute` for absolute delays.
Achievements and Future Work

- **Achievements**
  - Port and Test the Basic Primitives over FijiVM

- **Future Work**
  - Port the upper layer of the ESL to Java
  - Build a component based system as a simple example of use the Java ESL and FijiVM on LEON3
  - Adapting the automatic code generation to the Java ESL
  - Build a test bench to certify the whole product.
  - Build a Java prototype of the Nanosat on board software running on LEON