AIRBUS INTERNSHIP - STATIC ANALYSIS OF JAVA AVIONICS SOFTWARE

Online application available at:
http://www.eads.com/eads/france/fr/travailler-pour-EADS/postuler/rechercher-les-offres.html (FR)
or http://www.eads.com/eads/int/en/work-for-eads/apply/search-for-vacancies.html (EN)

Use reference 10229426 as a keyword to find this internship.

Department
Airbus Operations SAS (Toulouse) – avionics software

Supervisor
David Delmas – david.delmas@airbus.com

Education
Master 2 or Engineering School in Computer Science / Software / Real Time Software

Start date
3 February 2014 (or later…)

Duration
5-6 month

Technical skills
Extensive knowledge of the **Java** programming language.
Notions of compilation.
Prior exposure to **formal verification** techniques, though not compulsory, would be most helpful.
Notions of **programming language semantics** and **abstract interpretation** based static analysis would be ideal.

Description of the internship
"Avionics & Simulation Products" (EYY) is responsible for the design, development, production and support of part of the computers embedded on Airbus aircrafts. The "Verification and Dependability Support" group of the "Avionics Software" department (EYYW) is involved in transferring formal techniques from academic research into the verification processes of actual avionics software. Some of these techniques are already used industrially for the certification of several safety-critical avionics programs. The intern shall join the team and contribute to their R&T activities.

The internship aims at evaluating static program analysers for Java in an avionics context. The intern will study in particular Julia (juliasoft.com), an abstract interpretation based static analyser which aims at proving automatically certain safety and security properties of Java programs.

The intern shall familiarise himself with the scientific foundations of static analysis by abstract interpretation, and experiment some static analysers on avionics software developed at Airbus. Precise analyses (few or no false alarms) can be obtained through iterative refinement by the user, taking into account the abstractions computed by analysers. If some false alarms cannot be cleared by analysis refinement, the intern shall characterise the invariants abstracted too coarsely, in order to suggest prototype improvements to analyser developers.