Code Analysis of
Safety-Critical & Mission-Critical Systems Using ASIS

SIGAda'99 Extended Abstract

The Ada Semantic Interface Specification (ASIS) is an interface between an Ada environment as defined by ISO/IEC 8652 (the Ada95 Reference Manual) and any tool requiring information from this environment. An Ada environment includes valuable semantic and syntactic information useful for assessing software quality. ASIS-based tools can support code analysis to assure high quality systems both during initial code development and especially for independent verification and validation.

This type of code analysis is especially critical for embedded computer systems having mission-critical and safety-critical properties. This presentation focuses on the use of ASIS for such systems with applicability to the Safety and Security Guidelines being developed for ISO by the Safety and Security Rapporteur Group (HRG).

Examples of tools that benefit from the ASIS interface include: automated code monitors, browsers, call tree tools, code reformators, coding standards compliance tools, correctness verifiers, debuggers, dependency tree analysis tools, design tools, document generators, metrics tools, quality assessment tools, reverse engineering tools, re-engineering tools, style checkers, test tools, timing estimators, and translators. In fact most of these have already been developed using the ASIS interfaces by people in Australia, Canada, China, Denmark, France, Germany, Japan, Russia, Sweden, Switzerland, the United Kingdom, and the United States.

The ASIS 83 interface was developed by the Association for Computing Machinery (ACM's) Special Interest Group on Ada (SIGAda) through its volunteer effort in the ASIS Working Group (ASISWG). ASISWG has continued this important work for Ada 95 in conjunction with the ISO/IEC JTC1/SC22 WG9 ASIS Rapporteur Group (ASISRG) to standardize ASIS as an international standard for Ada 95. ASIS is now available as an International Standard denoted as:

ISO/IEC 15291:1999 Information technology — Programming languages
— Ada Semantic Interface Specification (ASIS)

The ASIS Standard is available via the ISO Catalog at http://www.iso.ch/infoe/catinfo.html; the ASIS specific reference is located at http://www.iso.ch/cate/d27169.html.

The ASIS 95 is rather mature. By SIGAda'99, there should be at least 4 implementations, AONIX, ASIS for GNAT, DDC-I, and Rational. A fifth may be available in the November 1999 timeframe. The ASIS for GNAT compiler was developed by Dr. Sergey Rybin from Moscow State University in conjunction with Professor Alfred Strohmeier of the Swiss

Federal Institute of Technology. This complete ASIS implementation is released to the public with GNAT.

The WG9 Safety and Security Rapporteur Group (HRG) see the value of using ASIS to analyze conformance of safety critical applications to their safety and security guidelines.

Examples of code analysis oriented towards embedded mission critical application will be provided. ASIS allows analysis for all the compilation units within the partition. This is quite useful for distributed embedded applications. There will also be a focus on how one might develop their own tools to analyze their own code.

The first ASIS-based tool for Ada 95 was demonstrated at STC'97. The tool is ObjectMaker from Mark V systems. Part of ObjectMaker included a syntactic parser and a semantic analyzer to generate a call tree. This original functionality took 6 1/2 man-years of effort. This functionality has now been replaced with ASIS95 calls, updating the tool for Ada 95. The replaced functionality took approximately 1 week of effort. This is an excellent demonstration of the power that ASIS brings to the Ada community.

The paper will focus on a newly developed template, which facilitates the implementation of an ASIS-based tool to perform code analysis. Examples using this template to perform code analysis in the following 3 areas will be provided:

1. Identification of Declarations for *Information Flow Analysis*
2. Call Tree for *Control Flow Analysis*
3. Restrictions Checker for *Formal Code Verification*

Discussion may be extended to address specific code analysis needs of the distribution and real-time community.

The ease of making ASIS-based tools allows code developers of embedded systems with a powerful capability to develop their own tools to support their own specific analysis needs.

KEYWORDS: ASIS, Ada, Ada 95, Code Analysis, IV&V, Software Testing, Reliability, Safety and Security, HRG, Mission-Critical, Safety-Critical