

**ASIS Workshop Report**  
**7-9 PM, Sunday, 12 November 2000**  
**Johns Hopkins University/Applied Physics Laboratory**  
**SIGAda 2000, Laurel, Maryland, USA**

The primary focus of the Ada Semantic Interface Specification (ASIS) Working Group and ASIS Rapporteur Group has been to evolve ASIS as an interface to the Ada 95 compilation environment. ASIS now provides a powerful mechanism to perform code analysis for mission-critical and safety-critical applications. A variety of highly effective tools have been built using ASIS. ASIS became an ISO standard in 1999.

An ASIS Workshop was held on Sunday, 12 November, as part of SIGAda 2000 in Laurel, Maryland to address several outstanding issues. These issues are:

1. ASIS Charter – Now that ASIS is an ISO standard, the ASIS Charter needs revision. This workshop proposed a revised charter to address our current focus.
2. Use of Electronic Mail Lists – There have been concerns over the use of the ASIS Technical maillist. Do we need an additional maillist?
3. Revision to ASIS for Ada 05 – WG9 is planning to have an evolved version of the Ada language circa 2005. How can we position the Toolkit Layer and Extensions Layer to support the evolution of ASIS?
4. Future Directions of ASIS Technology – What can we do to better position ASIS technology to our current/new users and future ASIS Standards?

Participants from the compiler vendor community and user community were present and provided their views and their needs. The following were participants in this ASIS Workshop:

Currie Colket, Workshop Chair, colket@colket.org  
Clyde Roby, Workshop Recorder, IDA, roby@ida.org  
David R. Phillips, Cohesion Force, david@cohesionforce.com  
Sergey Rybin, ACT, Rybin@gnat.com  
Alok Srivastava, TRW, alok.srivastava@trw.com

ASIS has had a presence at Ada-Europe 2000 and will have a presence at SIGAda 2000. The following papers were presented at Ada-Europe:

1. Rybin, S., A. Strohmeier, V. Fofanov, and A. Kuchumov, ASIS-for-GNAT: A Report of Practical Experiences, Lecture Notes in Computer Science, 1845, Reliable Software Technologies Ada-Europe 2000, Springer-Verlag, Berlin, pp. 125-137. This paper discusses the main difficulties met when implementing ASIS, the problems in using ASIS, and missing features (e.g., analysis of object-oriented programs). An Object-Oriented ASIS (OASIS) research project to develop an ASIS secondary library of interfaces is discussed. This report provides an excellent focal point for future activities to evolve ASIS when we eventually support the next version of the Ada programming language.
2. Hovater, S., W. Marksteiner, and A. Butturini, Generation of Interface Design Description Documents Using ASIS, Lecture Notes in Computer Science, 1845, Reliable Software Technologies Ada-Europe 2000, Springer-Verlag, Berlin, pp. 138-148. This paper discusses the automatic creation of Interface Design Description documentation can by using representation clauses and annotations for key record types and ASIS-based tools to extract this information from the Ada compilation environment.
3. Closhen, P. and H. Hoffmann, An Ada95 Programming Style Checker Using the GNAT ASIS Implementation, Lecture Notes in Computer Science, 1845, Reliable Software Technologies Ada-Europe 2000, Springer-Verlag, Berlin, pp. 149-160. An implementation of a style checking framework written in Ada95 and using GNAT ASIS is presented. The framework allows one to specify what is being presented.

The SIGAda 2000 Conference also has a number of ASIS related activities. These include:

1. ASIS Workshop -- 7:00-9:00 PM, Sunday, 12 November.
2. Tutorial – Building Ada Development Tools with ASIS-for-GNAT by Sergey Rybin, Monday, 15 November, 1:30-5:00 PM.
3. Invited Paper – Generation of Documentation using ASIS Tools by Steve Hovater, Wednesday, 15 November, 4:00-4:45 PM.
4. Invited Paper – Ada Code Analysis: Technology, Experience, and Issues by Dan Cooper, Thursday, 16 November, 9:30-10:15 AM, WG9 – Friday

These artifacts are published on the SIGAda 2000 Conference Proceedings CDROM.

## Charter

The ASIS Working Group is chartered under the Association for Computing Machinery (ACM) Special Interest Group on Ada (SIGAda). The SIGAda Executive Committee approved the initial ASISWG Charter on 8 February 1993. A revision was approved on 28 June 1994. The current charter was approved on 7 November 1995 at Tri-Ada'95. With the approval of ASIS as an ISO Standard as ISO/IEC 15291:1999 Information technology — Programming languages — Ada Semantic Interface Specification (ASIS), the charter is now out of date and requires revision. The current ASISWG Charter reads as:

### CHARTER OF ASISWG (Approved 7 November 1995)

*The purpose of this working group is to develop and participate in standardizing an implementation independent application programming interface to retrieve information from an Ada environment.*

*The Ada Semantic Interface Specification (ASIS) is an interface between an Ada environment (as defined by ISO/IEC 8652:1995) and any tool requiring information from it. An Ada environment includes valuable semantic and syntactic information. ASIS is an open and published callable interface which gives CASE tool and application developers access to this information. ASIS has been designed to be independent of underlying Ada environment implementations, thus supporting portability of software engineering tools while relieving tool developers from needing to understand the complexities of an Ada environment's proprietary internal representation.*

*The working group is currently developing ASIS for Ada 95, and is working towards its adoption as an international standard. The working group successfully developed ASIS for Ada 83, which is now complete and in the public domain.*

*Membership in the working group is open to any interested party. Members are responsible for their own expenses. Current members include representatives of Ada compiler implementors, CASE tool developers, application developers, and other interested users.*

After much discussion, the proposed ASISWG Charter reads as (changed areas in bold):

### CHARTER OF ASISWG (Proposed 12 November 2000)

*The purpose of this working group is to **evolve** an implementation independent application programming interface to retrieve information from an Ada environment.*

*The Ada Semantic Interface Specification (ASIS) is an interface between an Ada environment (as defined by ISO/IEC 8652:1995) and any tool requiring information from it. An Ada environment includes valuable semantic and syntactic information. ASIS is an open and published callable interface, which gives CASE tool and application developers access to this information. ASIS has been designed to be independent of*

*underlying Ada environment implementations, thus supporting portability of software engineering tools while relieving tool developers from needing to understand the complexities of an Ada environment's proprietary internal representation.*

***The working group cooperated with the ISO/IEC JTCl/SC22 WG9 ASIS Rapporteur Group (ASISRG) in adopting ASIS as an international standard. The ASISWG is now focused on evolving this interface to better satisfy the needs of the ASIS community and to proactively encourage the development of ASIS-based tools.***

*Membership in the working group is open to any interested party. Members are responsible for their own expenses. Current members include representatives of Ada compiler implementors, CASE tool developers, application developers, and other interested users.*

It was agreed that the proposed charter should be circulated amongst the ASIS electronic mail list for comments. The resulting document would be presented to the ACM SIGAda Executive Committee for approval.

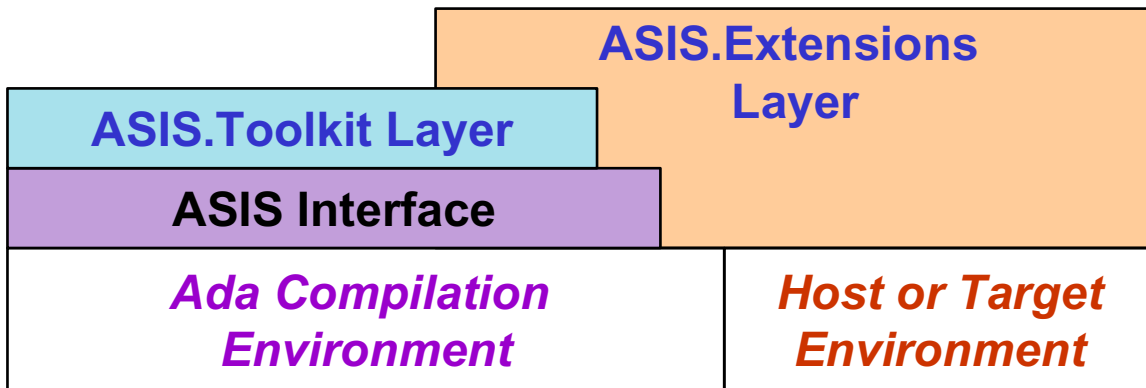
### **Additional ASIS Mailing Lists**

Currently the ASISWG two electronic mail forums: one for general ASIS information and one for technical discussions about ASIS itself:

***SIGAda-ASIS @ acm.org*** for ASIS high-level discussions  
***SIGAda-ASIS-Tech @ acm.org*** for ASIS technical discussions

In our recent history, ASIS technical had been used primarily to support the ASIS standardization discussions. Recently new (and experienced) ASIS users have been using the ASIS technical list to ask questions as to the best way to use ASIS to perform various types of analysis. The recommendation was to create a third list to separate ASIS standardization issues from ASIS usage issues. After much discussion, it was decided that since the standardization issues are currently low volume, that it might be best to use the ASIS technical list to support both standardization and Ada usage issues. It was further noted that frequently Ada usage issues provide guidance as to how the ASIS standard should evolve so the 2 areas do complement each other.

### **Revision to ASIS for Ada 05 (Approach using Toolkit and Extensions Layer)**



**Figure 1 Relationship of the Extensions and Toolkit Layers to the ASIS Interface**

WG9 is planning to have an evolved version of the Ada language circa 2005. How can we position the Toolkit Layer and Extensions Layer to support the evolution of ASIS? ASIS Toolkit/Extensions Layers was the primary topic of discussion at SIGAda'99 in the ASIS Workshop. ASIS developers could extend ASIS: through a **Toolkit Layer** and an **Extensions Layer**. The **Toolkit Layer** would provide higher level, abstract secondary queries built from ASIS primitives and other portable services. It would be portable to ASIS implementations provided by ASIS vendors.

Such queries would be valuable as examples to reduce the learning curve in using ASIS. The *Extensions Layer* would allow vendors to fill holes and gaps in the ASIS interface and support queries that ASIS was never intended to support such as dynamic semantics. Figure 1 depicts the relationship between the Toolkit Layer and the Extensions Layer to the ASIS Interface, the Ada Compilation Environment, and the Host or Target Environment. The Extensions Layer would be able to use queries from the Toolkit Layer, as these are portable. The converse is not true; should the Toolkit Layer use queries from the Extensions Layer, the Toolkit Layer would no longer be portable.

Even with the limitation of using ASIS interfaces, the Toolkit functionality is limitless. It would be able to support the following types of functionality in a portable way across all Ada vendors supporting the ASIS Standard:

- Initialize and Finalize the ASIS environment and context;
- Select compilation units using mechanisms such as filters, wild cards;
- Classify elements with new queries such as: Is\_Package, Is\_Task, Is\_Global;
- Fully classify types by cutting through derivations and private types;
- Develop queries to support Object Oriented code analysis such as Controlling\_Parameter, and Corresponding\_Tagged\_Type;
- Support the use of Lists, Trees, Sorting, and other utilities;
- Offer generic report templates;
- Build iterative structures like Call Trees and Dependency Graphs; and
- Perform source code transformations.

Functionality from the Extensions Layer would support an even greater level of capabilities not possible with the ASIS interfaces alone, such as:

- Provide sizes and alignments for types and objects;
- Provide record layout information such as: component positions, first\_bit, and width;
- Provide sizes for arrays and components;
- Provide frame sizes and stack offsets of local objects;
- Provide addresses of subprograms, tasks, or data within a target specific partition or program;
- Provide filename, directory, or other host dependent functions.

There are known weaknesses in the ASIS ISO Standard. The production of interfaces for the Toolkit Layer and Extensions Layer represent an excellent way to compensate for the current weaknesses and prepare ourselves for the next version of ASIS. Assuming that the next version of ASIS is going to build on the existing version of ASIS, then vendors will develop interfaces that will fall into either the Toolkit Layer or the Extensions Layer. Take for example the Object-Oriented ASIS secondary layer proposed in Dr. Rybin's Ada-Europe 2000 paper. This is being developed, as the existing ASIS interfaces are weak in their support for OO. It is obvious that an additional layer such as OASIS might contain interfaces in both the Toolkit and Extensions layer. We need to set up a process to periodically review these interfaces and formally identify potentially valuable interfaces for the next version of the ASIS interface. The user community will have the opportunity to use the interfaces and provide feedback thus improving the interfaces for the ASIS community. If these are made available to the international ASIS community to promote the portability of tools across platforms, then the ASISWG/ASISRG could recommend to WG9 those that are valuable for the next version of ASIS. This is a good time to set up the process to facilitate the generation of the next ASIS interface. This would expedite the approval of the next ASIS ISO Standard.

Please see the ASIS Workshop for SIGAda'99 for additional discussion to include proposed guidelines on how the two new child packages Asis.Toolkit and Asis.Extensions might be used.

### **Future Directions of ASIS Technology**

All agreed that we need more ASIS-based tools. There are probably about a dozen such tools, most provided via ACT and Rational. The major new player is DCS. DCS is currently marketing a tool to automatically analyze Ada95 source code, especially for functional metrics, object-oriented metrics, and ISO WG9 Safety and Security Rapporteur Group (HRG) restrictions checking. It is very encouraging to have a tool vendor, who is not also a

compiler provider, provide ASIS-based tools. In discussing this issue, it was recognized that there are likely many more ASIS-based tools internal to companies and vendor's organizations. One of the new roles identified in the charter is to *proactively encourage the development of ASIS-based tools*.

The question is how do we go about doing this? Porting tools from one compilation environment to another was a design goal of ASIS. Although differences in operating systems and compilation environments mean that this can not be achieved for most tools, it is hoped that tools can be developed that are extremely easy to port from one compiler provider's environment to another. Currently such a tool does not exist. The existence proof with a highly portable ASIS-based tool might be an excellent way to proactively encourage the development of ASIS-based tools. Perhaps it would be valuable to have an ASIS-based tool in the public domain that provides something extremely useful. Then we can address issues associated with porting the tool to the ACT, DDC-I, OC Systems, and Rational environments. This would go a long way in quantifying the effort to make a port. It would also provide us with knowledge that might help us reduce the porting effort when an ASIS for a future version of Ada is developed. Some candidate tools were discussed that would be useful to the Ada community. These included:

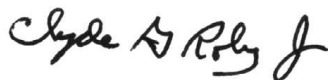
- Tool to eliminate the USE clauses in program by fully qualifying all references
- Tool to provide metrics on Ada statements
- Tool to provide robust control over the pretty printing of an Ada program

It was noted that not even the examples in Appendix B of the ASIS ISO Standard are fully portable across all vendor platforms. A useful tool that was portable explicitly demonstrating what is needed to make the tool satisfactorily execute in each environment was viewed as being extremely important to *proactively encourage the development of ASIS-based tools*. This was viewed as a goal we should achieve by next fall.

## Conclusion

The workshop recognized that ASISWG and ASISRG had done an excellent job in developing the ASIS standard. A proposed charter revision has been made to guide the ASISWG over the near future. The ASISWG will now explicitly continue its role to proactively encourage the development of ASIS-based tools. This will be of tremendous value to the Ada international community. The ASISWG/ASISRG will position itself to produce an ASIS ISO Standard for Ada 05. The Toolkit Layer and Extensions Layer should be valuable to the process of selecting candidate interfaces for the next version of ASIS. We would like to thank the Workshop participants for the valuable insight they provided.

Respectfully submitted,



Clyde Roby  
ASISWG Recorder



Currie Colket  
ASISWG Chair/ASISRG Chair