

**Workshop Report**  
**ASIS - Where Do We Go From Here?**  
**6-10 PM, Sunday, 8 November 1998**  
**SIGAda'98, Washington DC**

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. Balloting to approve ASIS as an ISO standard closed on 8 December 1998.

But, where do we go from here? An ASIS Workshop was held on Sunday, 8 November as part of SIGAda'98 in Washington D.C. to address this main question by addressing a number of issues. These issues are:

1. □ The ASIS standard provides for conforming extensions. The ASIS compiler vendor community is already providing extensions. **Would it be desirable to identify a secondary layer of such interfaces to support greater portability of ASIS programs? If so, which category of interfaces are needed; OO interfaces, library interfaces; tasking interfaces; run-time interfaces?**
2. □ A Program View Layer was provided for Ada 83. **Is such a layer desirable for ASIS for Ada 95?**
3. □ ASIS currently provides a number of mechanisms to address OO analysis. The last couple of interfaces added to the ASIS standard provide direct benefit to the OO community. **Would additional mechanisms be of value to the OO community?**
4. A number of tutorials exist on the ASIS Home Page (<http://www.acm.org/sigada/wg/asiswg>). **Are these useful to software engineers wanting to use ASIS? Would tutorials addressing a different focus be valuable?**
5. Currently there is no textbook on the bookshelves addressing the value of ASIS and how to use ASIS effectively. **Would such a book be desirable? Is this something we could author ourselves?**
6. Some have suggested a closer tie to the analysis of linked object code and even run-time debugging tools. **Would it be desirable to evolve extensions to facilitate such analysis? Or should we say we have completed our job, pat ourselves on the back and go home?**
7. Some have suggested that ASIS can be an effective mechanism for expediting the commercial use of Ada. **Can ASIS be an effective tool to support this effort?**
8. Some believe we need ASIS development tools to support the needs of the user community. **Is this an area the ASISWG and ASISRG should focus future efforts?**

Participants from the compiler vendor community, tool 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](mailto:colket@colket.org)  
Clyde Roby, Workshop Recorder, IDA, [roby@ida.org](mailto:roby@ida.org)  
Roy Bell, Raytheon, [rmbell@acm.org](mailto:rmbell@acm.org)  
Steve Blake, Aonix, [sblake@aonix.com](mailto:sblake@aonix.com)  
Ken Brooker, Corps of Engineers, [kbrooker@tec.army.mil](mailto:kbrooker@tec.army.mil)  
Jim Hurst, SofTools, [jshurst@oakland.edu](mailto:jshurst@oakland.edu)  
Robert Leif, Ada Med, [rleif@rleif.com](mailto:rleif@rleif.com)  
Joe Meyer, DoD  
Stephen Michell, Maurya Software, [steve@maurya.on.ca](mailto:steve@maurya.on.ca)  
Thomas A. Panfil, NSA, [Thomas\\_A\\_Panfil@acm.org](mailto:Thomas_A_Panfil@acm.org)  
David R. Phillips, Cohesion Force, [david@cohesionforce.com](mailto:david@cohesionforce.com)  
Ronald Price, McDonald Bradley, [rprice9979@aol.com](mailto:rprice9979@aol.com)  
Gil Prine, Unisys, [prine@mindspring.com](mailto:prine@mindspring.com)  
Sergey Rybin, ACT, [Rybin@gnat.com](mailto:Rybin@gnat.com)  
Paul Stachour, Univ. St. Thomas, [pdf@stachour.com](mailto:pdf@stachour.com)  
Bill Stanley, SofTools, [wfstanle@oakland.edu](mailto:wfstanle@oakland.edu)  
Alfred Strohmeier, EPFL, [alfred.strohmeier@epfl.ch](mailto:alfred.strohmeier@epfl.ch)  
Alok Srivastava, TRW, [alok.srivastava@trw.com](mailto:alok.srivastava@trw.com)  
Bill Thomas, MITRE, [bthomas@mitre.org](mailto:bthomas@mitre.org)

Before tackling the issues, there was concern raised that the ASIS Draft International Standard (DIS) had been removed from the ASIS Home Page. This is in accordance with ISO policy. Their interim policy allows a Committee Draft (CD) to be available via an electronic forum such as the World Wide Web, ftp, or CD ROM. However, once the standardization process passes into the DIS stage, the document can no longer be made available via a publicly accessible electronic forum. As ASISRG is a group chartered under ISO WG9, we must abide by the rules. Mr. Jim Moore, the Convenor of WG9 has directed us to make available the ASIS packages as these are needed by the ASIS community in an electronic format. Further, this does not compromise the ISO requirement of not making the standard available on line. The ASIS packages are available on the ASIS Home Page in ASCII and HTML. The ASIS standard is produced in both Microsoft Word and PDF. An editor's version of the ASIS standard can be made available to those wishing to review it, providing comments for future versions. Please contact Mr. Clyde Roby at [robby@ida.org](mailto:robby@ida.org) if you are willing to be a reviewer.

**Issue Discussion:** Several of the easy issues were addressed first, saving the harder ones for later.

**Issue #4:** A number of tutorials exist on the ASIS Home Page (<http://www.acm.org/sigada/wg/asiswg>). **Are these useful to software engineers wanting to use ASIS? Would tutorials addressing a different focus be valuable?**

Currently the tutorials available are written to address a "sophisticated" user. People assigned to work on ASIS tools are not always on this level. Mr. Dan Cooper (by email) had expressed the desire for a tutorial oriented towards someone who is at a lower entry level and not quite as "sophisticated." Mr. Ron Price raised the same concern, as he has had relatively new people, needing training, to work on ASIS products. Professor Strohmeier indicated that ASIS is a sophisticated tool and the current level of the tutorial on the ASIS Home Page is very appropriate. Mr. Steve Michell suggested that this is not the right group to address a lower-level tutorial. Perhaps this is something that the education community could do. We should encourage educators to teach ASIS in their courses and perhaps develop ASIS tutorials, which could be made publicly available via the ASIS Home Page. Dr. Sergey Rybin presented an ASIS tutorial at SIGAda'98. This recent tutorial is now available as a link through the ASIS Home Page.

It was concluded that ASISWG/ASISRG should encourage the education community to include ASIS in their courses and that the ASIS Home Page could be a site for making such courses available to other educators.

**Issue #5 -** Currently there is no textbook on the bookshelves addressing the value of ASIS and how to use ASIS effectively. **Would such a book be desirable? Is this something we could author ourselves?**

Robert Leif indicated that the desire for additional tutorials is really a desire for a textbook. Perhaps this is a training issue for people, with the assumption they have Ada95 knowledge. ASISWG could do it, or convince someone (a few people) to put together something for a textbook. This was not viewed as a viable commercial interest as there are not enough readers. Also, people just need a week or so of help to use ASIS and so a tutorial should be enough. A good set of tools would help ASIS more than a textbook. We are in the post-Gutenberg age; we don't need a hardcopy, in fact, electronic tutorials are better. We can put an electronic form of any ASIS tutorial information on the ASIS Home Page, and if desirable, into ACM's Digital Library.

It was viewed that a textbook would consume resources and could not be commercially viable to justify publishing. On the other hand, we need to encourage the development of additional tutorial and educational resources, which we can place on the ASIS Home Page.

**Issue #7 -** Some have suggested that ASIS can be an effective mechanism for expediting the commercial use of Ada. **Can ASIS be an effective tool to support this effort?**

This issue was placed on the agenda by Dr. Robert Leif to open our eyes to ASIS usage, expediting the commercial use of Ada. Basically, programs could use code from reuse libraries. An ASIS based tool could analyze applications to determine royalties due the developer of the reused code. He also suggested we need to build tools, such as spreadsheets or word processing tools with Ada as the macro language. He also suggested a tool, in ASIS, to read Ada environment information and generate XML. Dr. Robert Leif gave a very interesting and thought provoking presentation. Please see his Report on the "How Do We Expedite the Commercial Use of Ada?" Workshop held on Monday, November 9, at SIGAda'98 for additional details. It is available at the <http://www.acm.org/sigada/wg/cauwg> URL.

**Issue of Extensions - Issues #1 (conforming extensions), #2 (Program View Layer), #3 (Extensions for OO analysis), #6 (extensions for analysis of linked object code/run-time debugging), and #8 (ASIS tools)** were discussed together as they all are related to extensions of ASIS.

Extensions to the ASIS specification may be required for a host of valid reasons. These reasons include:

- **Provide higher level abstractions** to increase productivity and effectiveness of the ASIS tool developer. The ASIS interfaces reflect the low-level syntactic level of the source code. A higher level abstraction might incorporate a number of ASIS queries. Such higher level abstractions could be useful to different classes of tool developers to support specialized requirements (e.g., additional OO analysis, static run-time analysis). These interfaces should be implementable using the standard ASIS interfaces. For ASIS 83, the Program View Layer (PVL) was developed to provide such abstractions. The PVL was implemented in 100% ASIS 83 interfaces, thus making the secondary layer portable across ASIS implementations. Such a secondary layer has not been developed or defined for ASIS 95.
- **Provide access to post-compilation/non-static information** (e.g., linked object code, run-time information). Note: This is clearly outside the design goals of ASIS.
- **Provide a read/write capability into the Ada compilation environment.** This has been forbidden by ASIS due to the high possibility of corrupting the data integrity of the Ada compilation environment. There may be some development environments where a write capability could have a benefit. Mr. Steve Michell suggested a High Integrity application where it would be useful to write back into the Ada environment to support levels of safety-criticality, without changing the source code. Mr. Roy Bell suggested a tool to identify places in packages where the delay statement is used and to automatically remove it, eliminating overhead of the runtime system.

Someone raised the concern that by discussing extensions, doesn't this sound an alarm that ASIS is not mature and is not complete? This is hardly the case. The maturity of ASIS has already been demonstrated with the Ada Compiler Validation Capability (ACVC) using ASIS for GNAT. The ACVC was compiled. ASIS was used to traverse the resulting Ada Compilation Library. Using only ASIS queries, the source code of the ACVC was reconstructed without having any access to the original ACVC source code. The ASIS generated source code for the ACVC successfully compiled and its object code compared exactly to the object code for the original ACVC. This demonstrated that ASIS is syntactically complete and highly mature. ASIS currently has a robust set of semantic queries. Extensions simply make the ASIS interface more powerful and easier to use.

The ASIS specification will not change. Currently, it provides a mechanism for vendors to provide extensions for their customers and for application developers to use these extensions, if desired. Using a vendor specific extension reduces the possibility for porting tools to another vendor's environment, limiting the ubiquity of ASIS based tools. A model was discussed allowing for a framework to discuss a variety of extensions. This model is shown in Figure 1 and supports the above reasons for having extensions. Notions introduced in the model include:

- **ASIS Abstraction Extensions - Secondary Layer Extensions** would provide layered support for higher level abstractions using the existing ASIS interface. Hence extensions in this layer are highly portable from one vendor's Ada compilation environment to another.
- **Vendor Extensions - Vendor Extensions** would provide interfaces directly into the Ada compilation environment to support a wide variety of requirements including those outside the scope of ASIS (e.g., Read/Write capabilities, object code analysis). An application using Vendor Extensions is not likely to be portable to another vendor's Ada compilation environment.

(Note: there was no agreement on names for these two extensions at the Workshop).

The Model clearly identifies that there is no restriction on Vendor Extensions, these can be read/write if the vendor desires. Also, a vendor may be able to significantly improve the performance of a high level abstraction through direct access to the vendor's Ada compilation environment. This interface could be made available through the secondary layer as a portable implementation exists. Hence the overlap of the Secondary Layer on the Vendor Extensions.

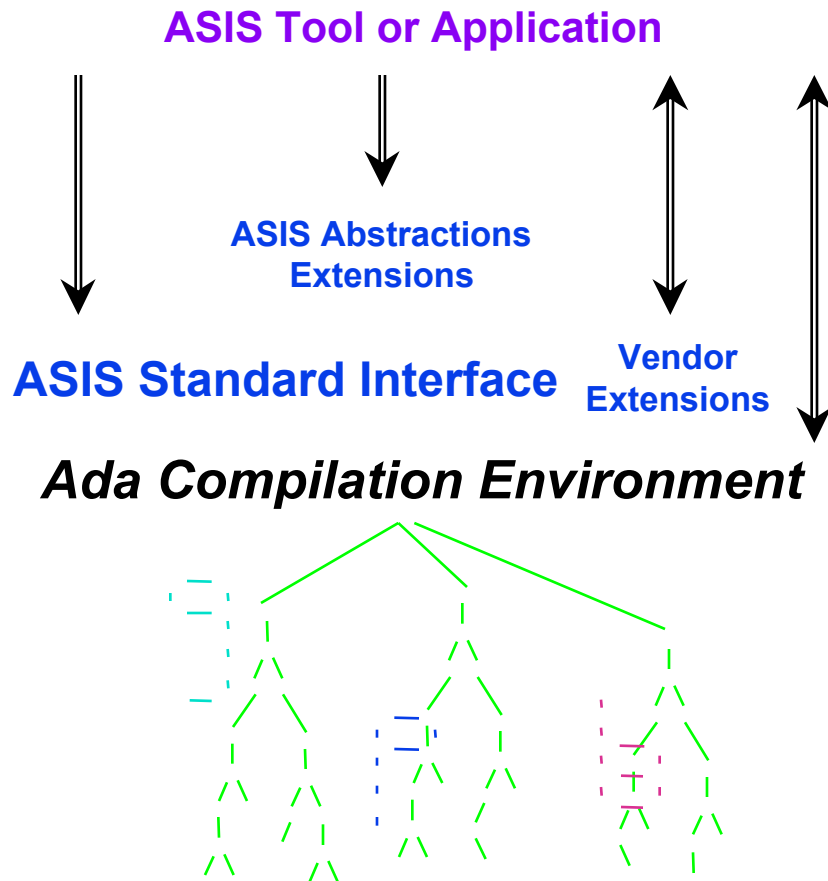


Figure 1 - Extensions to the ASIS Environment

**ASIS Abstractions Extensions (Secondary Layer)** - There was a considerable amount of discussion on this layer. The following questions were raised at the Workshop:

- **What is the Program View Layer (PVL)?** - The PVL is a widget-like layer of abstractions that model commonly used views of Ada program structure. Principle views included high level abstractions called:
  - **Region View** - identifies the declarative regions (i.e., symbol scopes) of an Ada library unit and organizes them into a tree structure to reflect their nesting relationships.
  - **Namespace View** - identifies the set of Ada entities (i.e., types, objects, subprograms, etc.) declared within an Ada unit.
  - **Reference View** - identifies the set of Ada entities referenced within an Ada unit. Each reference is classified with respect to its context, e.g., object references can be classified as 'read' or 'update'.
  - **Control Flow View** - identifies all possible transfers of control among the statements of an Ada program unit and represents them as a directed graph.
  - **Scanners** - provides a general purpose traversal template used by the views and ASIS applications.

Each view was implemented by a set of packages with operations for constructing and querying these high level views using ASIS queries. They offered tool builders the same kind of benefits enjoyed by X-users, such as productivity, consistent results, reliability, and portability. The PVL was developed by PRC for ASIS 83 as an Ada Technology Insertion Project. Dr. Bill Thomas indicated that MITRE found the PVL to be quite valuable in building Call Trees using the views provided. Many felt that the development of such a secondary layer is the most important task for the ASIS community. (Action: Mr. Clyde Roby took an action item to put the ASIS 83

PVL materials on the ASIS Home Page to facilitate everyone's understanding of the types of things we can do for ASIS 95. These are now available on the ASIS Home page, following the link to PVL under the ASIS Web Site Contents, to the [http://www.acm.org/sigada/WG/asiswg/ASIS\\_PVL.html](http://www.acm.org/sigada/WG/asiswg/ASIS_PVL.html) URL).

- **Would these abstractions be standardized?** - Since the abstractions should be portable to any ASIS 95 environment, there is no reason to standardize them. Also, ASIS is already an ISO standard. It is not desirable to change the standard for these extensions, even if it were possible.
- **Would the abstractions be managed?** - There was no clear consensus. ASISWG could take the lead on defining categories and layers of abstractions (there could be multiple layers). A small group within ASISWG could define an infrastructure of useful layers and abstractions. Source code for these abstractions could be donated to ASISWG as "shareware" and placed on the ASIS Home Page as "interesting stuff." There could also be another group of "very interesting stuff" that has gone through more consensus, more usage, more testing, etc. Perhaps artifacts falling in this last category should be published in Ada Letters to help establish the consensus. Every proposed extension should be useful for more than one ASIS tool. An alternative approach might be to have vendors build and distribute secondary layers. The ASISWG could disseminate information about companies who distribute secondary layer interfaces. There could be a matrix of "standard" secondary layer tools and who offers them. Specifications for abstractions should not be proprietary, as tool developers would need them to build applications. ASISWG could maintain vendor specifications for their abstractions on the ASIS Home Page. The implementations of these specifications may be proprietary. Hopefully developers of these high level abstractions will want to share the source code as well. These two approaches are not mutually exclusive and could work quite well together.
- **Implementation of Extensions?** - Child packages of ASIS packages are allowed. This could be an effective means to extend ASIS. The group viewed implementation of extensions using a child package of ASIS as something good. However, extensions should not be child packages of existing ASIS child packages.
- **Conventions for Extensions?** - A problem already exists where there are name collisions on ASIS extensions from the GNAT environment with ASIS extensions from the Aonix environment. The voluntary use of naming conventions by vendors could eliminate this problem. SIGAda could identify how extensions could be followed as a convention, which would benefit vendors and application developers alike. A "standard" naming convention would be valuable to all parties concerned. (Mr. Steve Blake volunteered to put a strawman on ASIS-Technical for further discussion; ACTION ITEM)
- **ASIS Consultants?** - Normally folks with ASIS questions simply send them out to ASIS Technical. Some one asked if it were possible to identify ASIS consultants to help with more involved tool building issues. Yes, this is possible. ASISWG currently maintains a list of vendors with ASIS products. Mr. Clyde Roby indicated that we could easily add those willing to consult to this list. Anyone wishing to be added to the list should send email to him at [roby@ida.org](mailto:roby@ida.org).
- **Could we have a Para-Secondary Layer?** - Mr. Gil Prine suggested a para-secondary layer as an alternative to having a read/write interface to the Ada Compilation Environment. Such a layer would use ASIS to recognize where changes are needed in the source code, open the source code file, make the change, recompile the source code, and use ASIS to verify the change is made. This is an interesting concept, but probably should not be part of the secondary layer.
- **Secondary Layer Tutorial?** - If categories and layers of abstractions were developed by ASISWG, then it might be useful to also develop a short tutorial on these abstractions, what they mean, and how to use them. This would be valuable regardless of how tools built on these abstractions are provided (via the WWW as "interesting stuff" or through tool vendors). This is obviously is a follow-on activity.

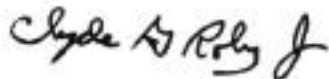
**Vendor Extensions** - There should be no limitation on Vendor Extensions. Secondary layer specification should be portable, including the implementation (as it is based upon ASIS). The "normative" implementation is one based on ASIS. However, vendor extensions, based upon a specific vendor's Ada Compilation Environment, do not even have to be based strictly upon ASIS. A vendor is free to develop such extensions to the benefit of the vendor and its customers.

**ASISWG Charter Revision** - During the discussions, the need for a revised ASISWG Charter was recognized. The existing charter is oriented towards supporting the standardization of ASIS. Now that ASIS will be an international standard shortly, the charter should be revised to reflect the new directions of the ASISWG as reflected in this workshop. A new charter will be developed for presentation to the next SIGAda Executive Committee Meeting at SIGAda'99.

**Conclusion:**

The workshop recognized that ASISWG and ASISRG had done an excellent job in developing the ASIS standard. This will be of tremendous value to the Ada international community. After standardization, these organizations will continue in their role of resolving issues affecting the standard. In addition, ASISWG has an important role in defining conventions for extensions and in defining a secondary layer to support the development and commercialization of higher level abstractions for ASIS. To support this, Steve Blake would develop a strawman for extension conventions and distribute it to ASIS Technical for comment; Clyde Roby would post the ASIS 83 PVL to the ASIS Home Page to facilitate understanding of a secondary layer based on commonly used viewed of Ada program structure. The ASISWG would then commence work to define a secondary layer to better support the needs of the Ada community. 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