A S I S

Ada Semantic Interface Specification

An Interface to the Ada 95 Environment

ASIS 95 Panel

Ada-Europe'96
Wednesday, 12 June 1996
14h00-15h30 and 16h30-18h00
Montreux, Switzerland

La specification de l'interface pour la semantique d'Ada

Co-chaired: Mr. Currie Colket, Chairman ASISWG/ASISRG
Dr. Sergey Rybin, Moscow State University.

Panelists: Mr. Pascal Leroy, Rational Software Corporation
Mr. Jesper Jørgensen, DDC-I
Mr. Bertrand Petitprez, Sema Group Grenoble
Mr. Clyde Roby, Institute for Defense Analyses
Mr. Tom Strelich, GRC
Mr. Bill Thomas, MITRE

A S I S — Ada Semantic Interface Specification

Overview of Sessions

ASIS 95 Panel

Ada-Europe'96

First Session (14h00-15h30) => What is ASIS?

- Overview of ASIS - Mr. Currie Colket (20 minutes)
- ASIS 95 Interface - Mr. Jesper Jørgensen (25 minutes)
- ASIS for GNAT - Dr. Sergey Rybin (25 minutes)
- ASIS Resources on the Internet - Mr. Clyde Roby (20 minutes)

Break

Second Session (16h30-18h00) => ASIS Usage Experience

- Rational Usage Experience - Mr. Pascal Leroy (15 minutes)
- MITRE Usage Experience - Mr. Bill Thomas (15 minutes)
- ASIS for GNAT Usage Experience - Mr. Bertrand Petitprez (15 minutes)
- Tool Vendor ASIS Usage Experience - Mr. Tom Strelich (15 minutes)
- Panel Discussion Questions (30 minutes)

ASIS

Ada Semantic Interface Specification

An Interface to the Ada 95 Compilation Environment

Overview of ASIS
ASIS 95 Panel

La specification de l'interface pour la semantique d'Ada

Mr. Currie Colket

Chair, ACM/SIGAda/ASIS Working Group (ASISWG)
Chair, ISO/IEC JTC1/SC22 WG9 ASIS Rapporteur Group (ASISRG)

Space and Naval Warfare Systems Command
SPAWAR PMW 133; 2451 Crystal Drive
Arlington, Virginia 22245-5200
Phone: (703) 602-1483; FAX: (703) 602-6805

colket@smtp-gw.spawar.navy.mil

12 June 1996
Ada-Europe'96

ASIS

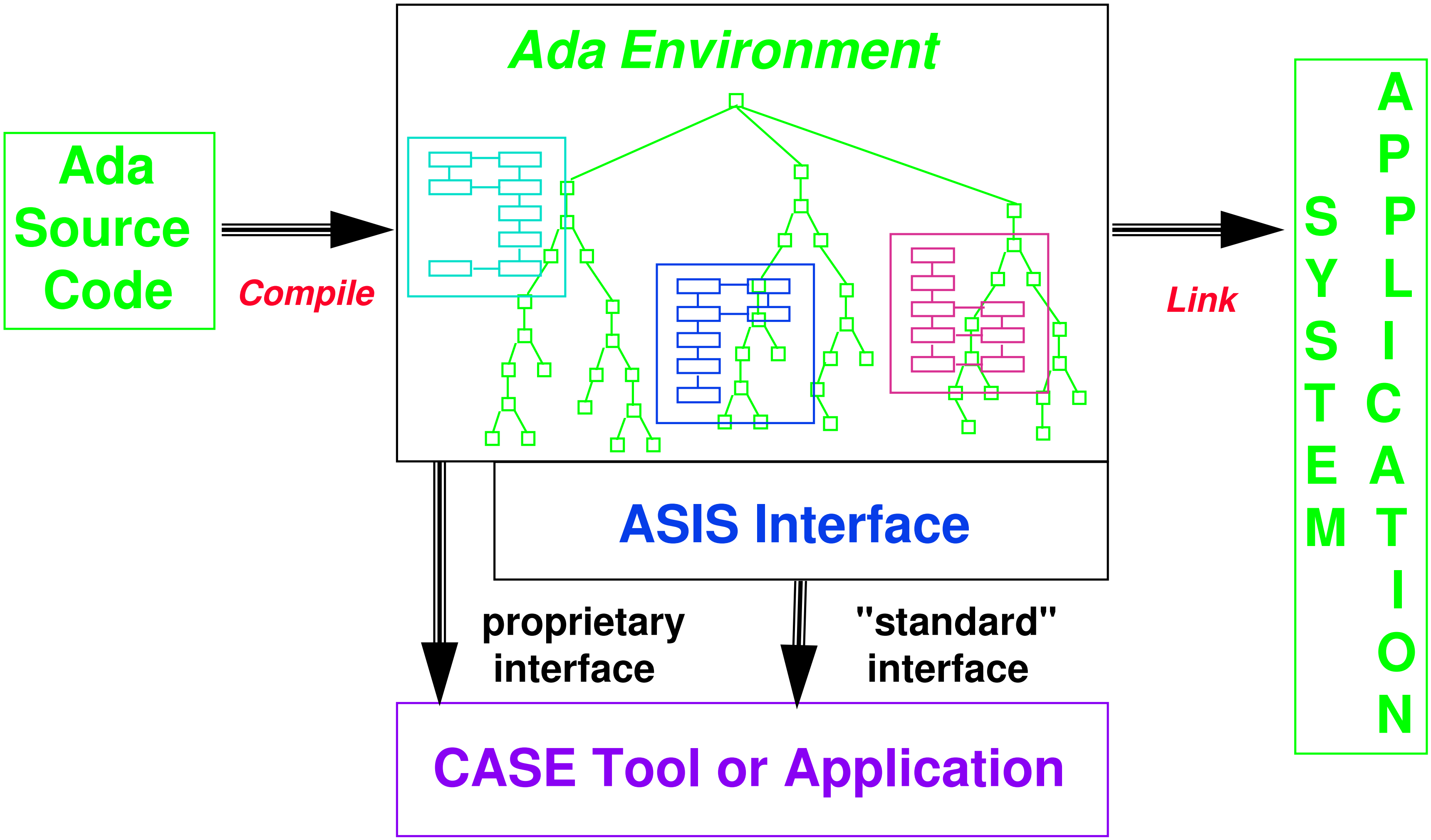
Ada
Semantic
Interface
Specification

An Interface to the Ada 95 Compilation Environment

Overview of ASIS

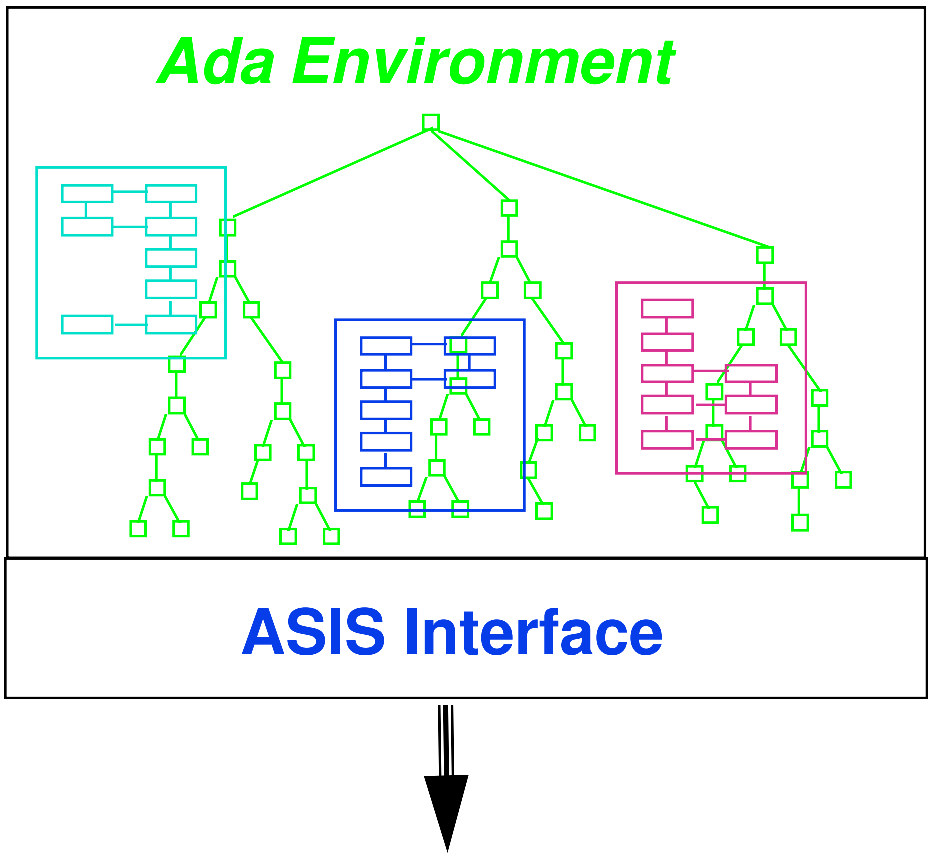
- **What is ASIS?**
  - **- Examples of Tools Built on ASIS**
  - **- Examples of Applications Built on ASIS**
- **ASIS Example**
- **ACM SIGAda's ASISWG**
- **ISO/IEC JTC1/SC22 WG9's ASISRG**
- **Expected ASISWG/ASISRG Schedule**

What is ASIS?



Provides Syntactic and Semantic Information from Ada Environment using a "standard" interface

Examples of Tools Built on ASIS



Tools portable to Ada environments supporting ASIS Interface

- Code Restructuring Tools
- Code Browsing and Navigation Tools
- Coding Style and Standards Compliance Tools
- Cross Reference Tools
- Data Flow Analysis Tools
- Dependency Tree Analysis Tools
- Design Tools
- Document Generation Tools
- Invocation (Call) Tree Analysis Tools
- Language-sensitive Editing and Prettyprinting Tools
- Language Translation Tools
- Quality Assessment Tools
- Reverse Engineering Tools
- Re-Engineering Tools
- Safety & Security Compliance Tools
- Static Correctness Verifiers
- Tasking Analysis Tools
- Test-case Generation & Coverage Analysis Tools
- Usage, Quality, & Complexity Metrics Tools

Safety and Security Compliance Tools

HRG (Annex H Rapporteur Group) Address safety and security issues by

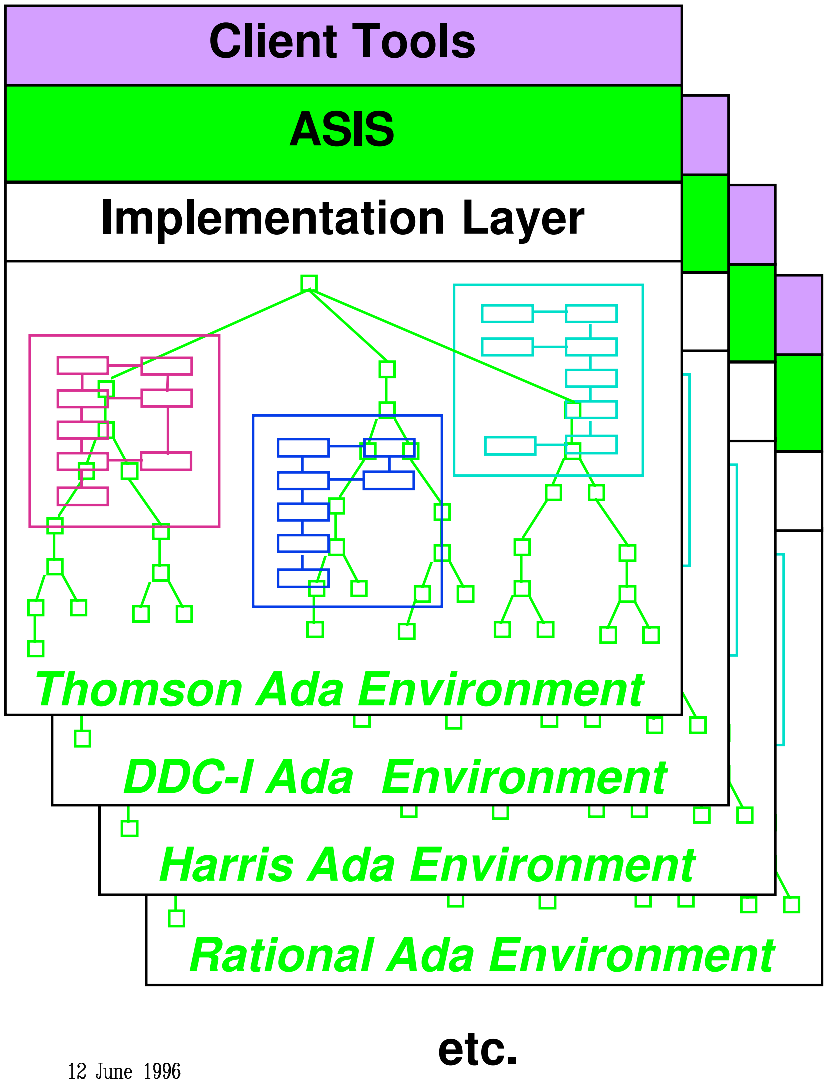
- Synthesizing essential requirements of typical sector-specific standards for high integrity applications tools
- Identifying restrictions for deterministic applications
- Producing taxonomy of techniques for the construction and analysis of high integrity software:
  - Use of annotations in program construction
  - Error detection by static analysis
  - Design confirmation by static analysis
  - Static timing analysis
- Supporting the interoperation of high integrity software with tools:
  - ASIS, Ada compilers and RTEs, & CORBA
  - Simple ASIS example to identify violations of restrictions might
    - » Scan your source code
      - Using ASIS Traverse_Element generic procedure
      - Actual for its Pre_Operation formal procedure should check the kind of the Element being traversed
    - » Report all occurrences of language constructs prohibited

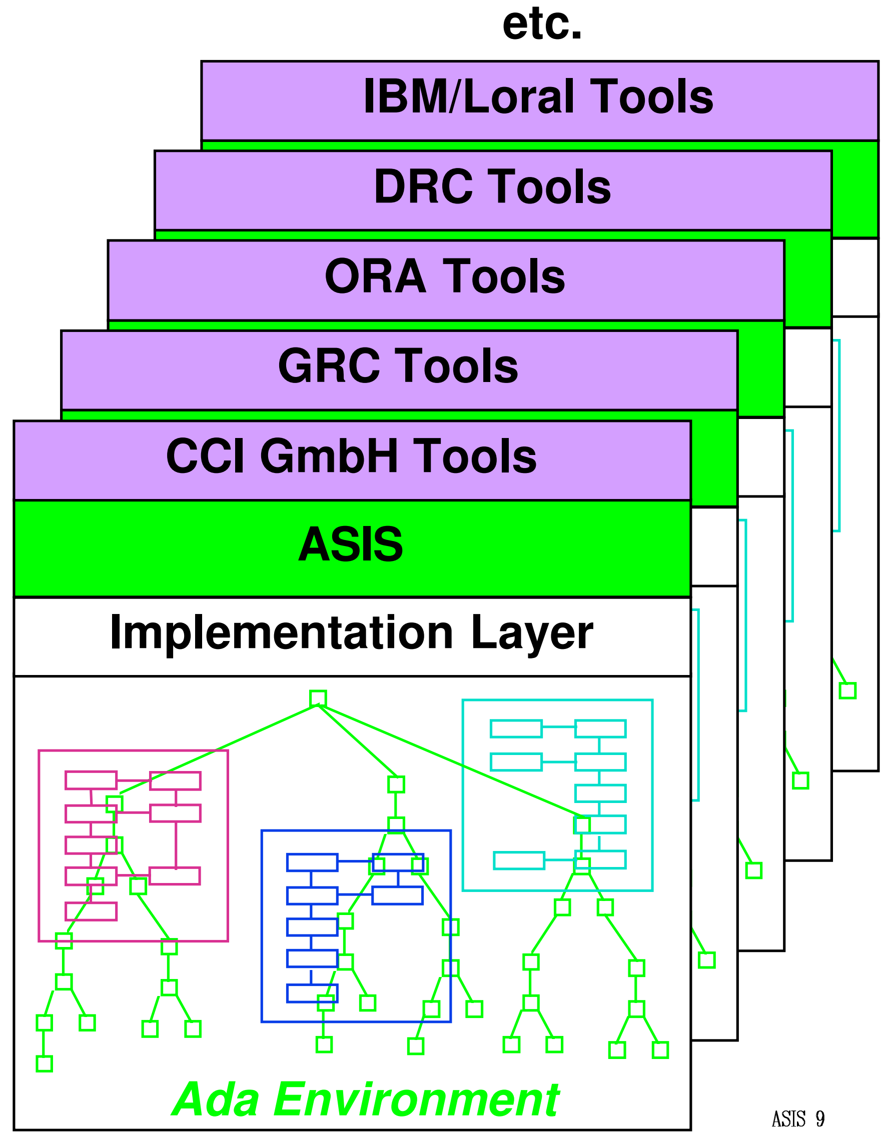
Safety and Security Compliance Tools

HRG (Annex H Rapporteur Group) Address safety and security issues by

- Synthesizing essential requirements of typical sector-specific standards for high integrity applications tools
- Identifying restrictions for deterministic applications
- Producing taxonomy of techniques for the construction and analysis of high integrity software:
  - Use of annotations in program construction
  - Error detection by static analysis
  - Design confirmation by static analysis
  - Static timing analysis
- Supporting the interoperation of high integrity software with tools:
  - ASIS, Ada compilers and RTEs, & CORBA
  - Simple ASIS example to identify violations of restrictions might
    - Scan your source code
      - Using ASIS Traverse_Element generic procedure
      - Actual for its Pre_Operation formal procedure should check the kind of the Element being traversed
    - Report all occurrences of language constructs prohibited

ASIS Lets Client Tools "Snap On" to Compilation Systems





ASIS Usage

Compiler Implementors:

- DDC-I *
- Harris *
- Intermetrics *
- OC Systems *
- Rational
- Thomson
- +
- FSF GNAT

Tool Vendors:

- CCI GmbH
- DRC
- Genoa Software *
- GRC
- Little Tree Consulting
- ORA Corp
- SoHar
- WPL Labs *
- Xinotech

End Users:

- Air Force
- Boeing
- IBM
- Logicon
- Loral
- Magnavox
- Martin-Marietta
- MITRE
- Navy
- Sema Group
- Unisys *
- +
- HRG

Countries:

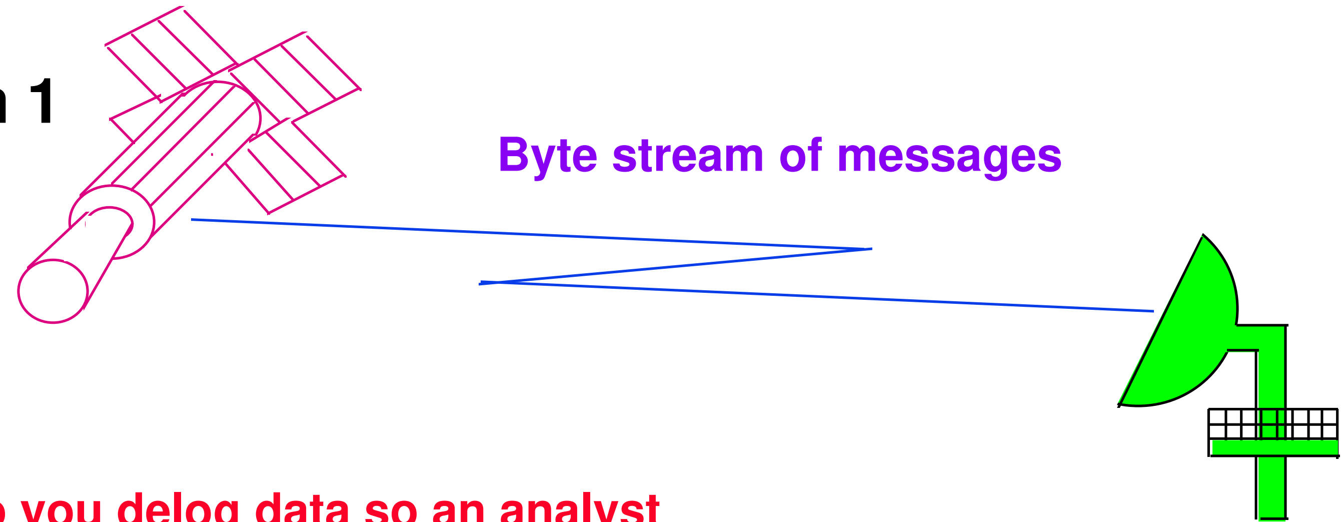
- Australia
- Belgium **
- Brazil **
- Canada
- Czech Republic **
- China
- Denmark
- Finland **
- France
- Germany*
- Japan*
- Korea **
- Netherlands **
- Russia
- Sweden
- Switzerland
- Ukraine **
- United Kingdom
- United States

* Using in near future

** Usage not known, but voted approval for ASIS at ISO/IEC JTC1/SC22 Ballot

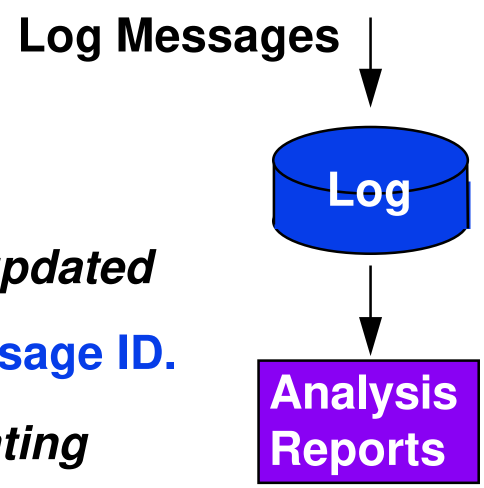
Example of Application Built on ASIS-1

System 1



Problem

How do you delog data so an analyst can view data in a meaningful way?



Alternatives

1. Delog routine per message ID.

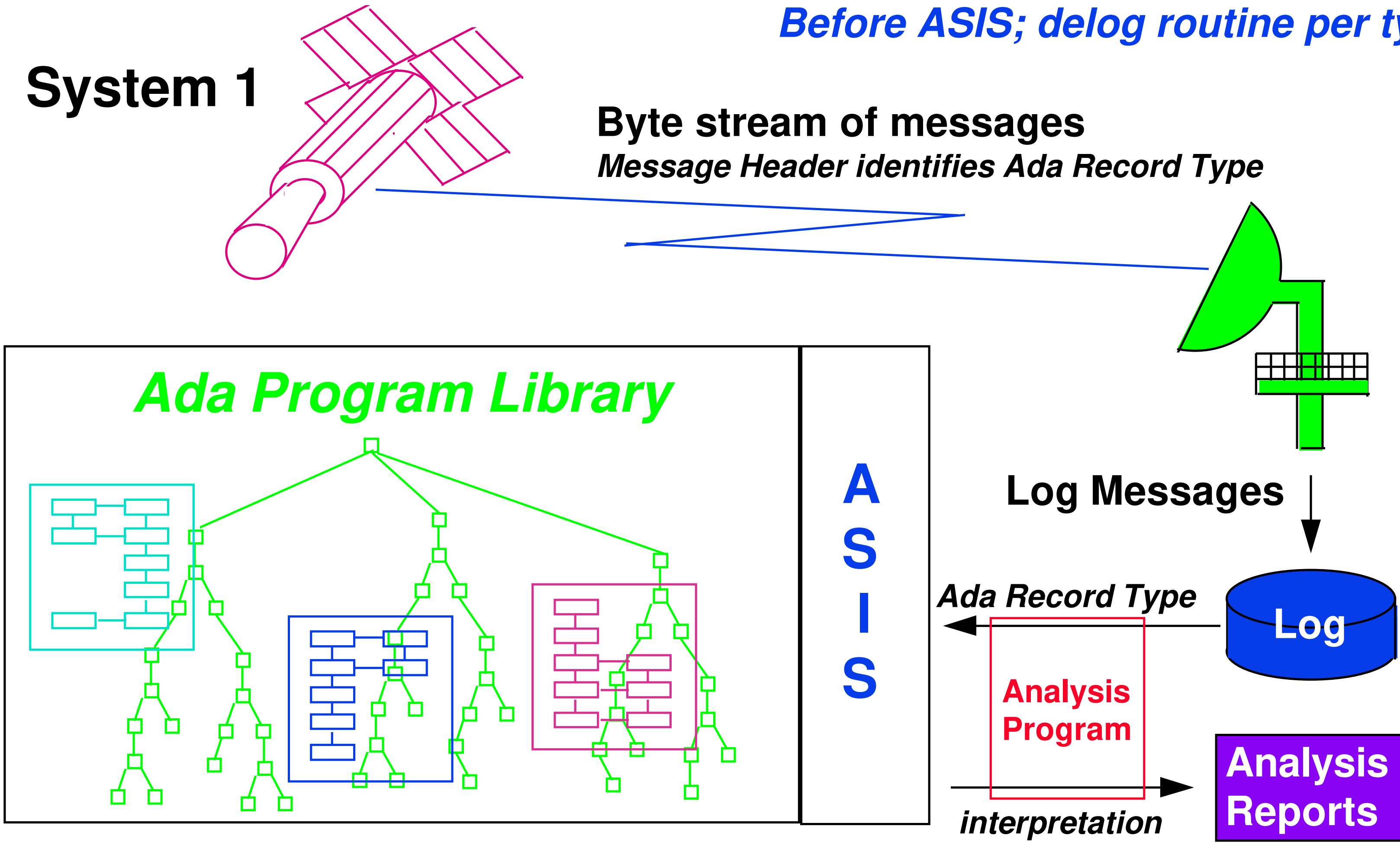
   Expensive, inflexible, must be constantly updated

2. Universal delogger, able to interpret any message ID.

   Effective, highly flexible, never needs updating

Example of Application Built on ASIS-2

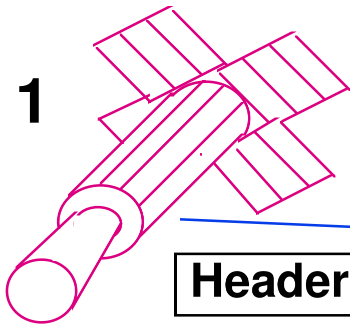
Before ASIS; delog routine per type



With ASIS; universal delogger, able to interpret any type;
Time reduced to build type dictionary from 704 hours to 2 hours

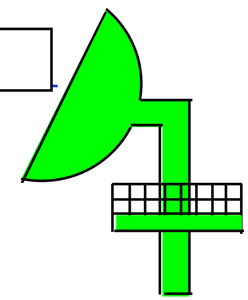
Example of Application Built on ASIS-3

System 1



Byte stream of messages
Message Header identifies Ada Record Type

Header	Byte_Stream (Array of Bytes)



```
      type message_1 is
          record
              field_1 : array_type;
              field_2 : record_type;
              field_3 : private_enumeration_type;
          end record;
      type message_2 (discrim_2: integer) is private;

  private
      subtype  range_type is positive range 10 .. 20;
      type varying_array_type is (integer range <> of float);
      type message_2  (discrim_2: integer);
          record
              field_1 : range_type;
              field_2 : varying_array_type (1 .. discrim_2);
          end record;
```

A S I S

Log Messages

Ada Record Type

Log

Analysis Program

Analysis Reports

interpretation services

ASIS provides 'size, 'position, 'first_bit, 'last_bit
Services index Byte_Stream to primitive types using Unchecked_Conversion
Services return values for analysis as variant record discriminated by ASIS Type_Kind

ASIS 95 Abstractions - Package ASIS

Ada Semantic Interface Specification (ASIS)

- **Provides queries & common types via 20 visible child packages**
  - Queries in form of function or procedure calls
  - ASIS Common types are implementation independent
- **Encapsulates vendor dependencies in:**
  - **ASIS_Vendor_Primitives**

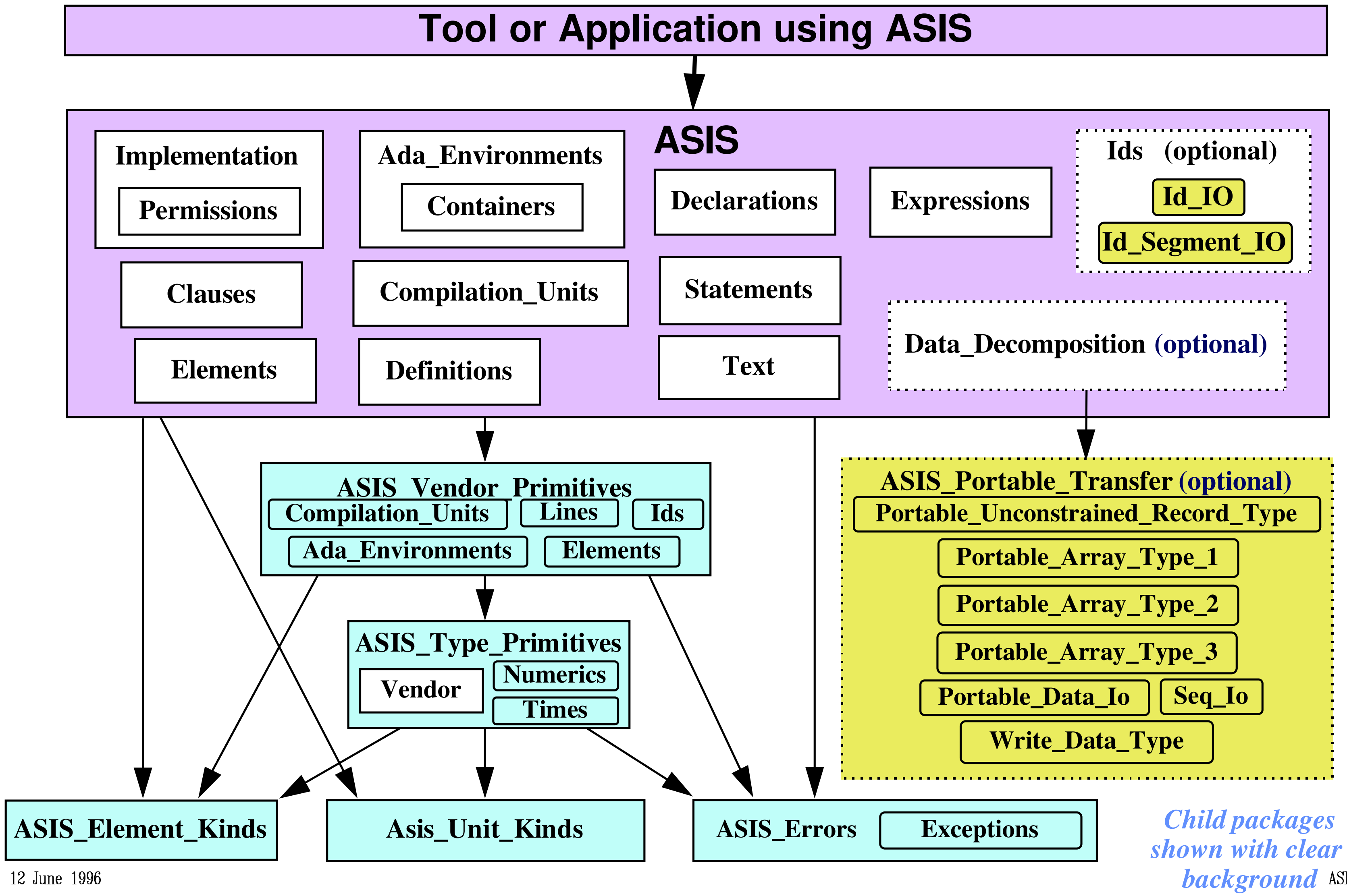
    Vendor dependent names and types are mapped to ASIS declarations
  - **ASIS_Types_Primitives**

    Visibility is controlled by subtype declarations and renaming declarations exposing only vendor independent operations

Designed to be portable for all implementations

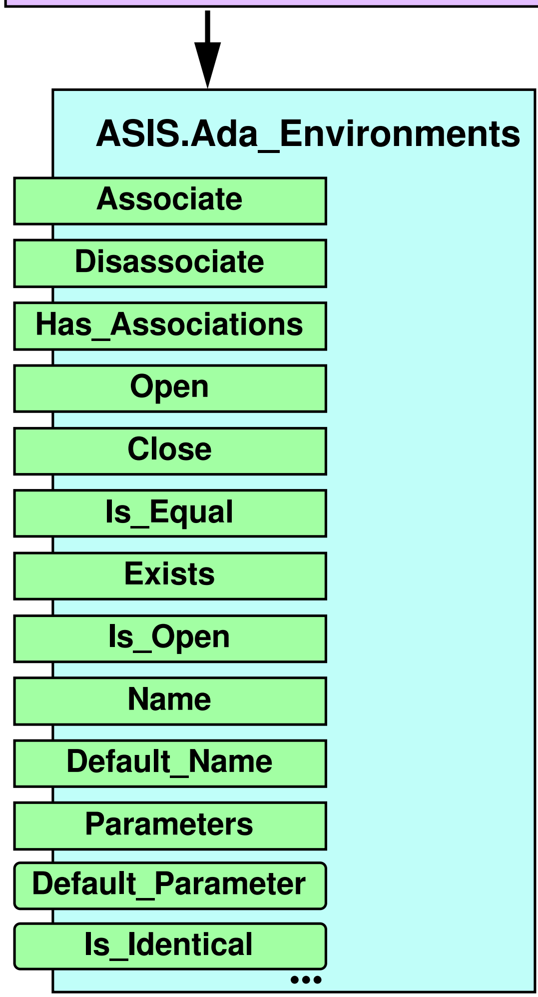
- **Error conditions result in Ada Exceptions**
  - **Status & Diagnostic information available**

ASIS 95 Package Architecture



ASIS 95 Abstractions - Ada_Environments

Ada Semantic Interface Specification (ASIS)



ASIS Context identifies an Ada Environment as defined by ISO/IEC 8652:1995 [Ada 95] which allows implementations to define methods to enter Compilation Units into the environment

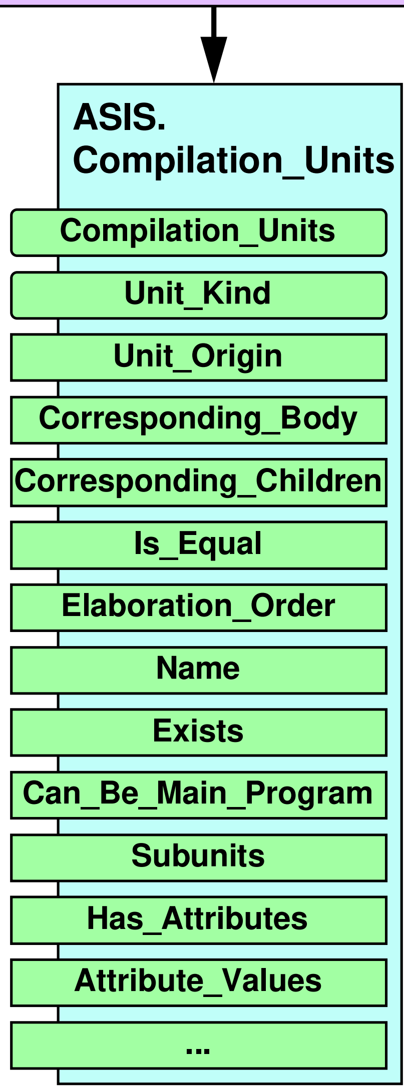
Environment Model:

- Associates name and parameters
- Open, Query, and Close
- Support for source-based, persistent, or multiple persistent libraries
- Free of vendor details and assumptions

In ASIS 83 ASIS.Ada_Environments was called ASIS_Libraries

ASIS 95 Abstractions - Compilation Units

Ada Semantic Interface Specification (ASIS)



Provides external "Black Box" view

- **Attributes:**

  Date compiled, Text File, Compilation Options, Optimizations, Can Be a Main Program

- **Relationships:**

  Supporters, Dependents, Closure, Implicit Dependencies

Compilation Unit Model
- **Fetch by name**
- **Query of attributes & relationships**
- **Gateway to internal view using Elements**

ASIS 95 Abstractions - Elements

Ada Semantic Interface Specification (ASIS)

...

Asis.Elements
...
Asis.Clauses
...
Asis.Declarations
...
Asis.Definitions
...
Asis.Expressions
...
Asis.Statements
...
Asis.Exceptions
...
Asis.Unit_Kinds
...
Asis.Text
...
Asis.Ids
...
Asis.Data_ Decomposition (Optional)

Provides internal “White Box” view

- **Logical handle to Ada elements:**

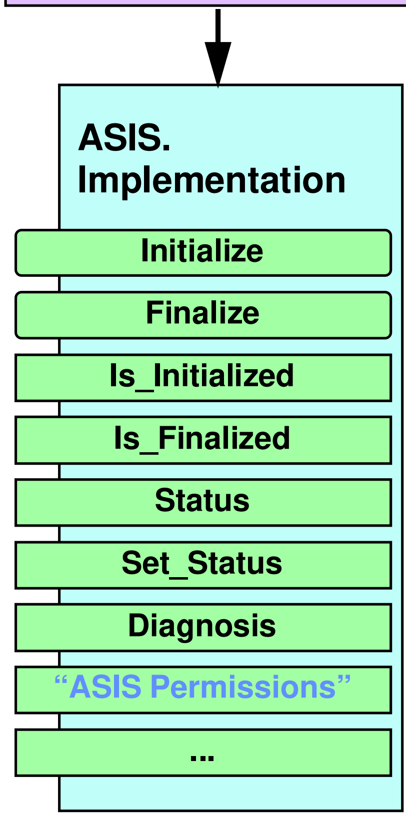
  Declarations, Statements, Expressions, Type Definitions, With Clauses ...

Element Model

- **Element kinds (Asis.Element_Kinds)**
- **Syntactical queries to classify, decompose, and recompose**
- **Semantic queries both forward & backward**
- **Elements know their enclosing Compilation Unit & Context**

ASIS 95 Abstractions - Implementation

Ada Semantic Interface Specification (ASIS)



Using ASIS:

```
Asis.Implementation.Initialize;                 -- To setup environment
...
-- Determine ASIS implementation permissions

P1 := Asis.Implementation.Permissions.Is_Line_Number_Supported;
P2 := Asis.Implementation.Permissions.Is_Commentary_Supported;
...
Asis.Ada_Environments.Associate (...);  -- To name Ada context
Asis.Ada_Environments.Open (...);        -- To gain (library) access
...
 -- Use various ASIS interfaces
...
Asis.Ada_Environments.Close (...);       -- To close (library) access

Asis.Ada_Environments.Disassociate (...);  -- To release context

Asis.Implementation.Finalize;                   -- To release all resources
```

Very Simple ASIS 95 Example - 1

```
with Text_IO; use Text_IO;
with Asis;
Procedure Asis_Application_Example is

      My_Context             : Asis.Context;
      My_Unit                : Asis.Compilation_Unit;
      Unit_Name              : String ( 1 .. 100 );
      Unit_Name_Length       : Natural;
      My_Control             : Asis.Elements.Traverse_Control
                                 := Asis.Elements.Continue;
      My_State               : Boolean := True;

   procedure Report_Declarations (Unit : in Asis.Compilation_Unit)
   ...

begin -- Asis_Application_Example

   Asis.Implementation.Initialize;
   Asis.Ada_Environments.Associate(My_Context, "My_Context");
   Asis.Ada_Environments.Open ( My_Context );

   Put_Line ("Type the name of an Ada package specification");
   Get_Line (Unit_Name, Unit_Name_Length);

   My_Unit := Asis.Compilation_Units.Library_Unit_Declaration
              ( Unit_Name ( 1 .. Unit_Name_Length), My_Context );

   if Asis.Compilation_Units.Is_Nil ( My_Unit )
   then
      Put ("Context does not contain the requested unit ");
   else
      Put ("Context contains the requested unit ");
      Report_Declarations ( My_Unit );
   end if;

   Asis.Ada_Environments.Close( My_Context );
   Asis.Ada_Environments.Dissociate( My_Context );
   Asis.Implementation.Finalize;

end  Asis_Application_Example;
```

```
For =>

package asis_test is

  type T is ( A, B, C);

  S : integer := T'BASE'SIZE ;

end asis_test ;
```

Simplified Example
to ASIS 2.0.G

```
Result =>

Type the name of an Ada package specification
asis_test
Context contains the requested unit
asis_test (is kind) A_PACKAGE_DECLARATION
T (is kind) AN_ORDINARY_TYPE_DECLARATION
A (is kind) AN_ENUMERATION_LITERAL_SPECIFICATION
B (is kind) AN_ENUMERATION_LITERAL_SPECIFICATION
C (is kind) AN_ENUMERATION_LITERAL_SPECIFICATION
S (is kind) A_VARIABLE_DECLARATION
```

Very Simple ASIS 95 Example - 2

```
procedure Report_Declarations (Unit : in Asis.Compilation_Unit) is

      My_Element    : Asis.Element;

    procedure Process_Element
                    (Elem : in Asis.Element;
                     Control : in out Asis.Elements.Traverse_Control;
                     State : in out boolean);

      procedure No_op
                    (Elem : in Asis.Element;
                     Control : in out Asis.Elements.Traverse_Control;
                     State : in out boolean);


      procedure Find_and_Print_Declarations is new
                     Asis.Elements.Traverse_Element
                         (Boolean, Process_Element, No_Op);

      procedure No_op
                    (Elem : in Asis.Element;
                     Control : in out Asis.Elements.Traverse_Control;
                     State : in out boolean) is
      begin
         null;
      end;

      procedure Process_Element ...

begin -- Report_Declarations

         My_Element    := Asis.Elements.Unit_Declaration ( Unit );
         Find_and_Print_Declarations (My_Element, My_Control, My_State);

end Report_Declarations;
```

Very Simple ASIS 95 Example - 3

```
procedure  Process_Element
               (Elem : in Asis.Element;
                Control : in out Asis.Elements.Traverse_Control;
                State : in out boolean) is

    package Kind_Io is new Text_Io.Enumeration_Io  (Asis_Element_Kinds.Declaration_Kinds);

   begin -- Process_Element

    case Decl_Kind is

       when Asis_Element_Kinds.Not_A_Declaration => null;

      when others =>

         if not Asis_Element_Kinds."=" (Asis.Elements.Declaration_Origin (Elem),
                Asis_Element_Kinds.An_Explicit_Declaration)  then  return;
        end if;

        declare

            Name_List : Asis.Defining_Name_List  := Asis.Declarations.Names (Elem);

        begin

            for I in Name_List'Range loop
                Text_Io.Put  (Asis.Declarations.Defining_Name_Image   (Name_List  (I)));
                Text_Io.Put ("  (is kind)  ");
                Kind_Io.Put  (Decl_Kind);
                Text_Io.New_Line;
           end loop;
        end;

    end case;

end  Process_Element;
```

For =>

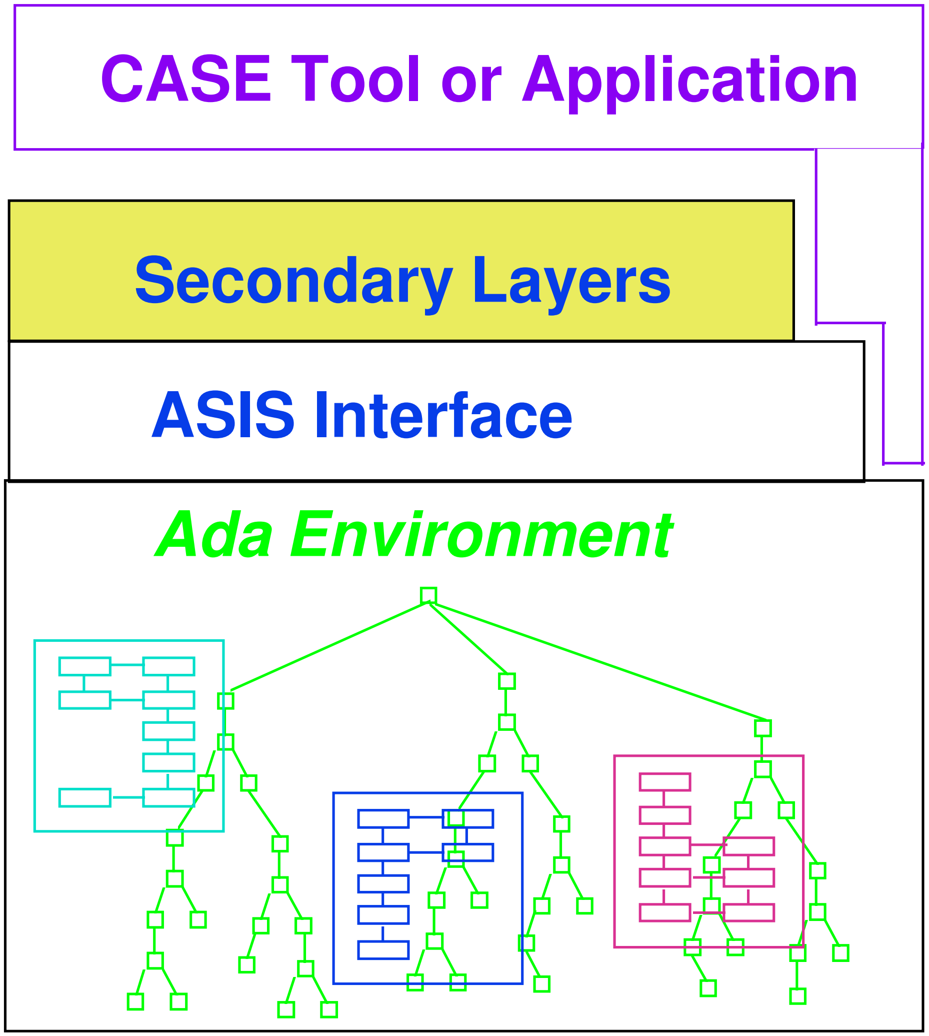
```
package asis_test is

   type T is ( A, B, C);

   S : integer := T'BASE'SIZE ;

end asis_test ;
```

Result =>

```
Type the name of an Ada package specification
asis_test
Context contains the requested unit
asis_test (is kind) A_PACKAGE_DECLARATION
T (is kind) AN_ORDINARY_TYPE_DECLARATION
A (is kind) AN_ENUMERATION_LITERAL_SPECIFICATION
B (is kind) AN_ENUMERATION_LITERAL_SPECIFICATION
C (is kind) AN_ENUMERATION_LITERAL_SPECIFICATION
S (is kind) A_VARIABLE_DECLARATION
```

ASIS Secondary Layers

Formerly called Program View Layers



- **Widget -like layer of abstractions that model commonly used views of Ada program structure**
- **Each view provided in set of packages with operations for constructing and querying views**
- **Offer CASE tool builders same kind of benefits enjoyed by X-users:**

  Productivity
  Consistent results
  Reliability
  Portability

- **Secondary layers available for:**

  Scanners
  Namespace View
  Reference View
  Region View
  Control Flow View

ASIS Ada Board Resolution

Motion at the Ada Board 10-11 September 1992:

> **The Ada Board recognizes the potential benefits to the DoD Ada community of an ASIS standard and therefore it recommends that the AJPO director support by whatever means possible, the development of an ASIS standard and its submission to ISO/WG9 for publication.**

Vote: Passed unanimously

ACM WG established to develop ASIS83 and ASIS95 with ISO WG9 Rapporteur Group for standardization of ASIS95

ACM's SIGAda's ASIS Working Group Charter

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.

Original February 8, 1993, Updated June 28, 1994, Updated November 7, 1995

ASISRG-1

ISO/IEC JTC1/SC22 WG9 ***voted unanimously to create ASISRG on 28 April 1995***

Scope

Standardize the interface between an Ada 95 compilation environment and tools/applications requiring information from this environment.

Purpose and Justification

The Ada Semantic Interface Specification (ASIS) is an interface between an Ada environment as defined by ISO/IEC 8652 and any tool requiring information from this environment. 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 having to understand the complexities of an Ada environment's proprietary internal representation.

ASISRG-2

Purpose and Justification (Continued)

As an international standard, ASIS will benefit the Information Technology community by facilitating the development of powerful CASE tools portable amongst the various environments provided by Ada vendors. This portability can only be achieved through the standardization of ASIS at the international level. A standardized ASIS will promote the development of powerful tools for the software engineering environment by providing access to important semantic information otherwise available only through proprietary interfaces. Further, ASIS will benefit the Information Technology community as a valuable resource for application development (e.g., decoupling system to system interfaces). The international standardization of ASIS will facilitate the use of this important capability in the development of system software applications.

ISO/IEC JTC1/SC22 approved ASISRG New Work Item on February 1996 with 17 nations voting for approval

ASISWG/ASISRG Officers

Chair ASISWG/ASISRG:	Currie Colket (SPAWAR)
Vice-chair ASISWG:	Steve Blake (Thomson)
Recorder ASISWG:	Clyde Roby (IDA)
Vice-recorder ASISWG:	Dan Cooper (Boeing)
Publicity/Meetings ASISWG:	Bill Thomas (MITRE)
ASISRG Co-Project Editors:	Steve Blake (Thomson) Clyde Roby (IDA)

How to Get ASIS - ASIS Artifacts

sw-eng host on Internet as *sw-eng.falls-church.va.us*

ASIS directories:

As of 19 April 1996

public/AdaIC/work-grp/asiswg
- README
- ASIS-FAQ
- ASISWG-Charter
- ASISRG-NWI
- ASIS Slides
- ASIS Tutorials
- meetings/
  - minutes/highlights/etc.
- asis/
  - bibliography/
  - issues/
  - V1.1.0/
  - V1.1.1/
    - ASIS 83 files
    - testing/
  - V2.0/
    - ASIS 95 files

Mirrored in other media
- AdaIC BB (modem)
- CD ROM
- Other US Internet nodes
- European Internet nodes

Call +1 (703) 681-2466 for help in accessing ajpo host

ASIS83 is complete and ready for use (postscript & ASCII)

ASIS 95 is under construction

ASIS Conformance Suite

Needed as a way to enhance tool-builder confidence that a particular ASIS implementation actually follows the standard

Valuable to vendor to ensure compliance

Suite of compliance tests is maintained with V1.1.1

- Complete with test harness
- ~ 1200 tests
- ~ 100 Mbytes
- Complete coverage for ASIS 83

  (except for module decomposition)
- Input from users for tests important to user community

ASIS GNAT Interface Implementation

GNAT is an Ada95 compiler developed by NYU and distributed by the Free Software Foundation

- **ASIS interface to GNAT is being developed by:**
  - **Dr. Sergey Rybin (Moscow State University)**

    in conjunction with
  - **Professor Alfred Strohmeier (Swiss Federal Institute of Technology)**
- **First free ASIS 95 implementation prototype available in February 1996**
- **Partial implementation; to be viewed as ASIS 95 prototype**

  Contains most of ASIS 2.0.E structural queries for GNAT 3.03 compiler
- **Currently limited to processing only one Compilation Unit at a time**

  A summer release should process all Compilation Units in the current directory

Available on WWW: http://lglwww.epfl.ch/cgi-bin/home page.p1 and/or ftp lglftp.epfl.ch in the /pub/ASIS directory

How to get ASIS - WWW, BB, Internet

- ACM World Wide Web

  URL => http://www.acm.org/sigada/WG/asiswg/asiswg.html

- Ada Information Clearinghouse Bulletin Board:

  By modem auto-baud to 14,400: +1 (703) 681-2845

  Call +1 (703) 681-2466 for AdaIC BB voice help

  Name => *guest* Password: *Ada4sw-eng*

  Parity = Even, Data Bits = 7, Stop Bits = 1, Xon/Xoff flowcontrol

- Ada Information Clearinghouse Internet via *ftp*:

UNIX> ftp sw-eng.falls-church.va.us	- Use ftp to sw-eng host
Name: anonymous	- Use *anonymous* login
Password: <user ID>	- Use user ID as password
ftp> cd public/AdaIC/work-grp/asiswg	- Change to asiswg sub-directory
ftp> ls	- List files in sub-directory
ftp> get README	- Retrieve asis README file

Call +1 (703) 681-2466 for AdaIC Internet voice help

Expected ASIS 95 Schedule

Dec 93	AJPO recommends ASIS V1.1.0 (ASIS83) be used as interface to Ada 83 Program Library
Mar 94	Design Goals for ASIS95 identified
Jun 94	ASISWG finalizes ASIS 83 as V1.1.1 with test suite
Jun 94	Evaluate design approaches for ASIS 95
Nov 94	Finalize approach for ASIS 95
Mar 95	Skeleton ASIS 95
28 Apr 95	ASISRG created unanimously by ISO/IEC JTC1/SC22 WG9
Jun 95	Skeleton ASIS 95; new library/environment model
Nov 95	Public Review of ASIS 95 initiated (Version 2.0.E)
2 Feb 96	ISO/IEC JTC1/SC22 approved NWI
Feb 96	ASIS available for GNAT Ada 95 Compiler
Mar 96	ASISRG vote to submit ASIS 95 as ISO Committee Draft
Jun 96	Expect ISO/IEC JTC1 approval of NWI - ASIS assigned ISO number
Fall 96	Possible ISO/IEC/ JTC1/SC22 WG9 approval for ASIS 95 CD
Sum 97	Possible ISO/IEC Registration as Draft International Standard
Fall 97	Possible ISO/IEC/ JTC1/SC22 WG9 approval for ASIS 95 DIS
Fall 98	Possible ASIS 95 approved as ISO Standard

Schedule viewed by many as overly optimistic
ASISRG => WG9 => SC22 => JTC1 cycle is slow

ASIS Interface - Funding

FY88 & FY89 Funding provided by STARS

Later in FY89, STARS initiated a policy not to fund any standardization efforts and ASIS funding was halted along with all other standardization activities

There has been no STARS, no AJPO, and no DoD funding for ASIS Standardization since FY89

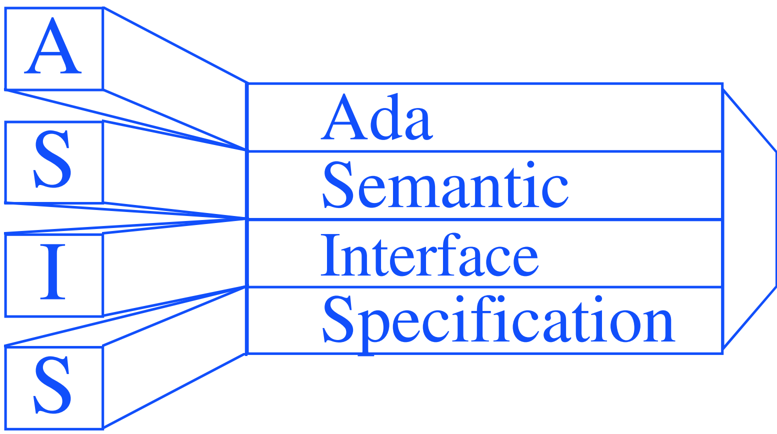
All post FY89 ASIS standardization funding has resulted from industry investment - user community working closely with the compiler implementor community

Final Notes on ASIS

ASIS - Interface for obtaining information from Ada environments

- Provides both syntactic and semantic information
- Supports building powerful and highly portable CASE tools
- Supports many requirements for building systems
- Used in approximately 10 countries
- ASIS83 mature; evolving ASIS95, useful today; standard ~ 98-99

For ASIS W W W URL => http://www.acm.org/sigada/WG/asiswg/asiswg.html



To get involved with ASISWG and/or ASISRG, send email to:

asis-request@sw-eng.falls-church.va.us

to join asis@sw-eng.falls-church.va.us

asis-technical-request@sw-eng.falls-church.va.us

to join asis-technical@ sw-eng.falls-church.va.us

or call Currie Colket @ +1 (703) 602-1483

or Clyde Roby @ +1 (703) 845-6666