ASIS/PVL

Ada Semantic Interface Specification/Program View Layer

Presented to:

Ada-Europe'96/ASIS 95 Panel

Presented by:

Tom Strelich
GRC International, Inc..
5383 Hollister Avenue
Santa Barbara, CA 93111
805/964-7724
tstrelich@sb.grci.com

GRC
INTERNATIONAL INC

Briefing Outline

- Introduction
  - Application area
  - ASIS experience
- Project Overview
  - Problem background
  - Technical Approach
  - Architecture
- View Descriptions
  - Control Flow View, Region View, Namespace View, Reference View
  - Scanner
- Productivity Improvements

Application Area: Software Metric Tools

AdaQuest™

New **Improved**

Software measurement and quality analysis tool

- **Coding Standards Audit –** checks code for conformance to the Software Productivity Consortium's Ada Quality and Style Guide V 2.01
- **Quality Metrics –** assesses reliability, maintainability, and portability based on the Rome Lab Software Quality Framework
- **Complexity Metrics –** computes cyclomatic complexity, identifies violations of user-specified goals and illustrates distribution of program complexity
- **Size Metrics –** provides detailed breakdown of physical and logical source statements
- **Development Environment Integration –** tightly integrated with the Rational Apex™ Ada development environment via ***ASIS***

GRC INTERNATIONAL INC

ASIS User/Implementor Experience

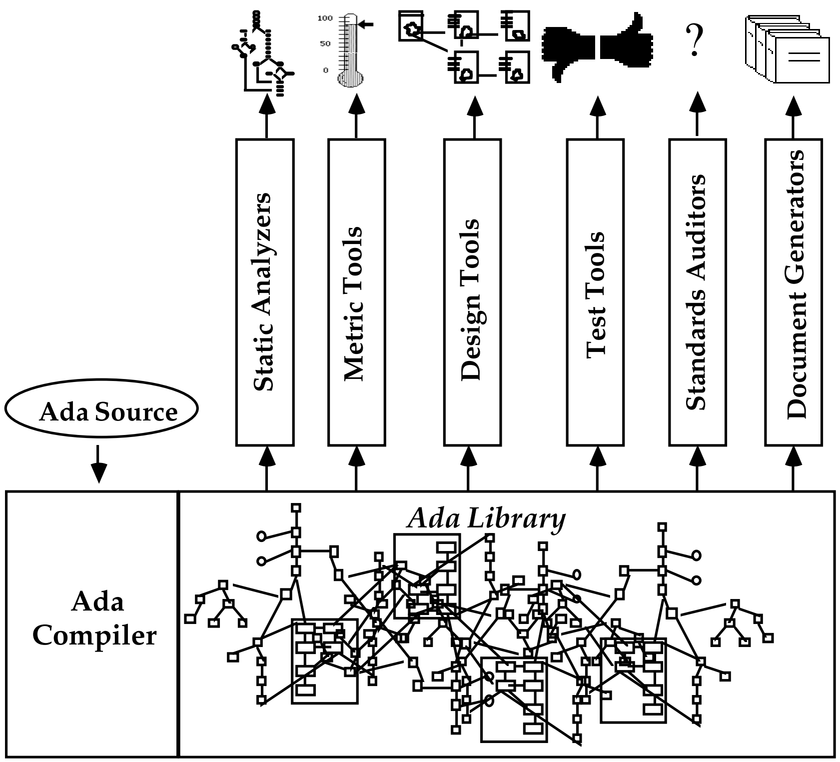
- Original AdaQuest product (Version 1.1) hosted on VAX/VMS with bundled Ada compiler front-end
- Migrated to Unix using ASIS in 1992
  - Problem: There was no VADS ASIS in 1992
  - Solution: We built our own
- ASIS Implementor
  - Built subset implementation (55%) of ASIS 1.0 on Verdix DIANA Library Interface
- ASIS User
  - Built AdaQuest 2.0 on ASIS
  - First commercial tool product built on ASIS (12/92)
  - Migrated to ASIS 1.1
  - Ported from Verdix/ASIS to Rational Apex/ASIS

GRC INTERNATIONAL INC

Problem Background

Advanced Ada tools have a common requirement:

They all need detailed semantic representations of the Ada code

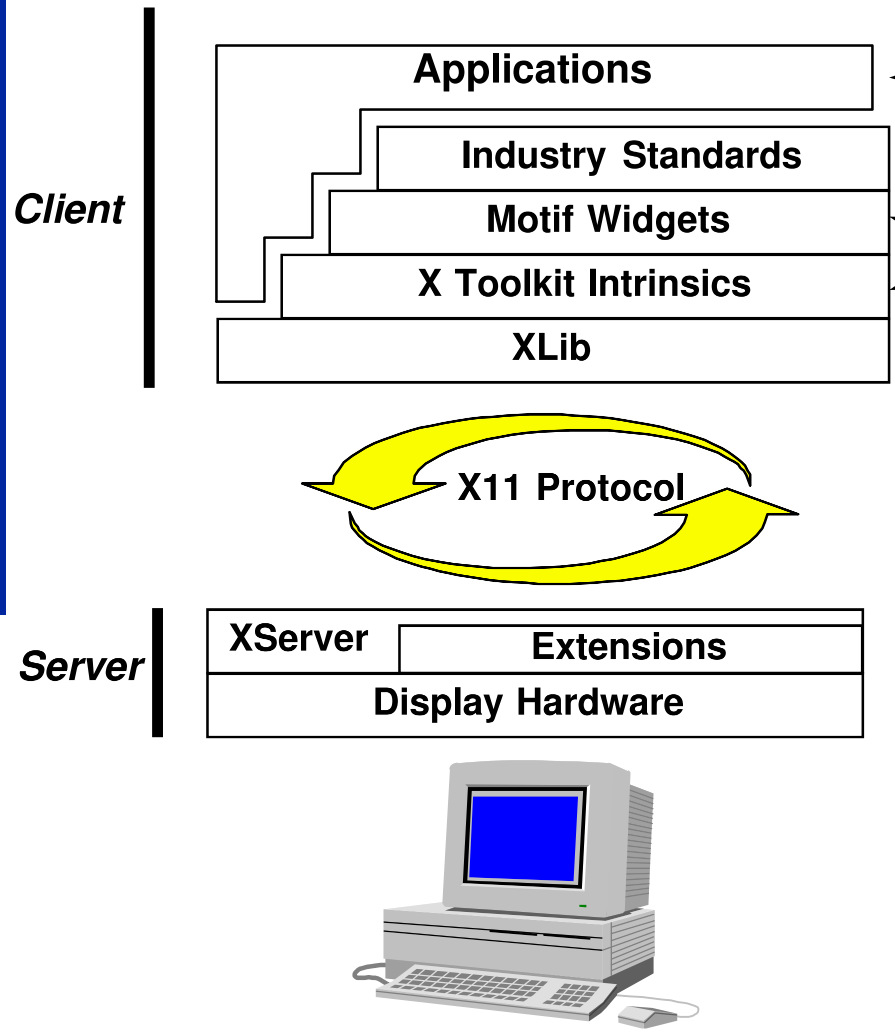


- **The Good News**
  - **ASIS provides tools with a portable interface to this semantic information**
- **The Bad News**
  - **ASIS models semantics at a very low level of abstraction...**

    58 different ASIS queries must be used to determine the set of program units called by a given program unit
- **The Need**
  - **Tool builders require abstractions of ASIS information that focus on different aspects of program structure**
- **We call these abstractions ...** ***Views***

X Window System Analogy

A layered set of programming interfaces for the graphical applications builder



Typical application builders use the higher-level layers almost exclusively

The XToolkit and widget layers implement higher level abstractions like menus, buttons and scroll bars in terms of these primitives

Xlib is the foundation layer, providing primitive services such as drawing lines on a workstation screen and handling keyboard/mouse inputs

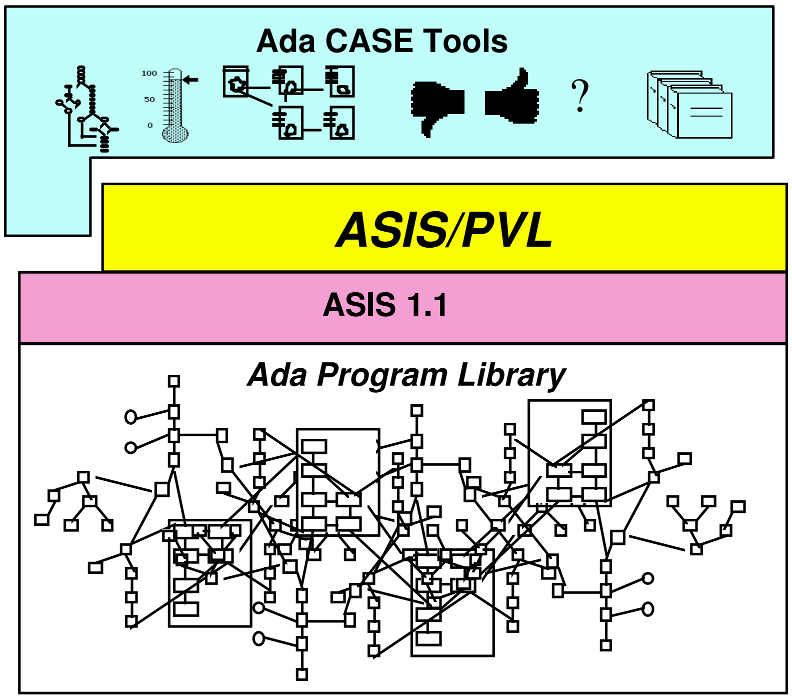
X Window System Benefits

- *Improves developer productivity*
- *Enforces commonality across applications*
- *Increases reliability (through reuse of "thoroughly punished" code)*

GRC INTERNATIONAL INC

ASIS Program View Layer (PVL)

ASIS/PVL is a widget-like layer of abstractions that model commonly-used views of Ada program structure



ASIS/PVL contains a set of Ada packages providing operations for constructing and querying views.

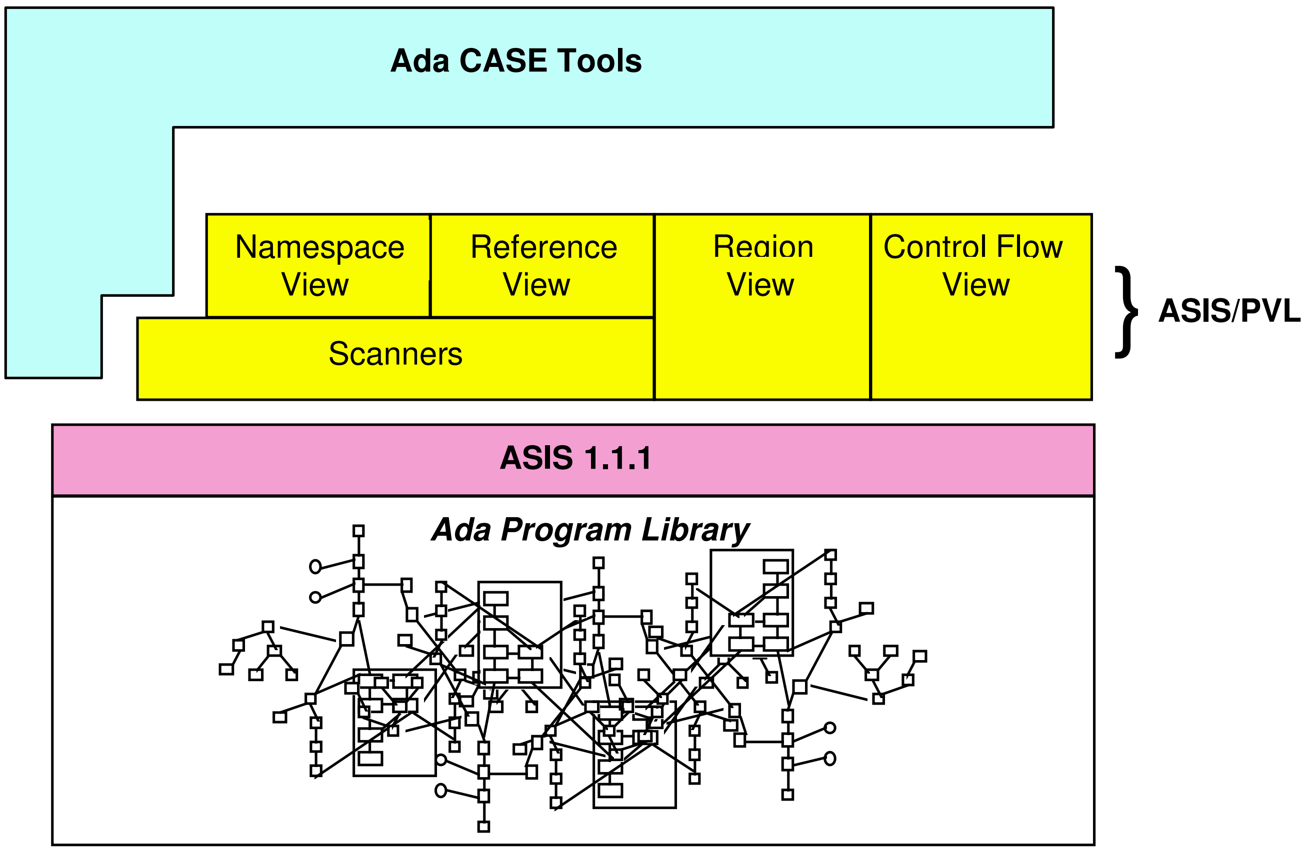
Each view is constructed from Ada program code using ASIS 1.1 queries.

ASIS/PVL offers Ada CASE tool builders the same kinds of benefits enjoyed by X users:

- **Productivity**
- **Consistency**
- **Reliability**

GRC INTERNATIONAL INC

ASIS/PVL Architecture



View Descriptions

- **Control Flow View**
  - Identifies all possible transfers of control among program unit statements represented as a directed graph
  - Utility: complexity measurement, test data selection, visual display of program structure
- **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
  - Utility: documenting the internal structure of a program unit, constructing a hierarchical symbol table(in conjunction with the Namespace view), distinguishing local vs. global symbol references in a program unit, determining the set of declarations declared by a library unit that are visible outside the unit
- **Scanner**
  - Decouples ASIS traversal from View construction and provides a generalized copy-edit template so that traversal can examine a variety of elements and preserve context when traversing ASIS element hierarchies: similar to ASIS generic "`traverse.element`" but more flexible and able to show the traversal path
  - Utility: when the application (e.g., tool) wants to consider a lot of different element kinds and when context dictates different operations on an element (e.g,, classify references such as set/use)
- **Namespace View**
  - Identifies the set of Ada entities (e.g., type, object, subprogram, task, etc.) declared in an Ada unit
  - Utility: constructing a hierarchical symbol table(in conjunction with the Region view), data dictionary construction, overload checking
- **Reference View**
  - Identifies the set of Ada entities (e.g., type, object, subprogram, task, etc.) referenced in an Ada unit
  - Utility: identifying unnecessary "with" clauses, identifying global variables, data flow analysis, identifying imports from other packages

GRC INTERNATIONAL INC

Productivity Improvements

- **AdaQuest 2.0 porting experience**
  - **ASIS vs DIANA**
    - » **2:1 productivity improvement**
    - » **Doing a given AdaQuest 2.0 static analysis in ASIS took half the development time as the equivalent analysis took using DIANA in AdaQuest 1.1**
- **AdaQuest 2.2 enhancement experience**
  - **ASIS/PVL vs ASIS**
    - » **5:1 to 10:1 productivity improvement depending on the nature of the analysis**
    - » **Use of the control flow view to compute cyclomatic complexity > 10:1 productivity improvement**
    - » **Use of the reference view to identify parameters used as control variables < 5:1 productivity improvement**