ASIS for
Analysis of Recorded Data

Presented at

Software Technology Conference

(STC'93)

on 20 April 1993

by Mr. Mickey White

of IBM FSC at Boulder, Colorado

As the original electronic version of this slide set was lost over the years,
this version was reconstructed from the hardcopy in April 1995.
An original hardcopy can be obtained from the STC Conference Management at +1 (801) 777-7411.
Please see the companion paper for this presentation, available through the ASIS Home Page at the
http://www.acm.org/sigada/WG/asiswg/asiswg.html URL.
Attachments 1, 2, & 3 of that paper are provided with this presentation as slides 15, 16, & 17.

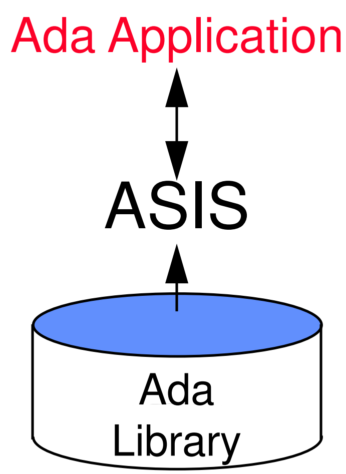
Overview

- What is ASIS
- ASIS versus Other Library Interfaces
- Applications Suited for ASIS
- How ASIS is Used
- History of ASIS
- IBM’s Use of ASIS - Recorded Data Analysis
- Standardization of ASIS - Status
- ASIS Working Group Information

What is ASIS

"ASIS defines a vendor independent interface for accessing both the syntactic and semantic information stored within an Ada compiler's library for an Ada compilation unit, including the information relating independently compiled units."

- An application programming interface (API)
- Binding to the Ada compiler library
- Set of Ada packages to be "withed" by your application
- Allows your application to query the Ada compiler library



ASIS Versus Other Library Interfaces

- Other library interfaces
  - DIANA
  - Rational LRM interfaces
- ASIS abstracts the Ada library structure
- Standardization of ASIS will facilitate portable applications

Applications Suited for ASIS

- Development environment tools - traversal
  Space_Craft: Space_Craft_Manager.Space_Craft_ID_Type;
- Reverse engineering tools
  - Dependency diagrams ("withing" structure)
  - Calling diagrams
  - Data dictionary
- Metrics tools
  - Ada9X compatibility analysis/conversion
- Document generation tools - hypertext
- Debuggers
- RECORDED DATA ANALYSIS

How ASIS Is Used

The semantics of ASIS are based on the Ada LRM Appendix E Syntax Summary.

- compilation_unit ::= context_clause library_unit
- library_unit ::= package_declaration | ...
- package_declaration ::= package_specification;
- package_specification ::= package identifier is {basic_declarative_item} private
  {basic_declarative_item} end [package_simple_name];

Given **Space_Craft: Space_Craft_Manager.Space_Craft_ID_Type**

Retrieve the declaration for Space_Craft_ID_Type.

How ASIS is Used - Simplified Example

- COMPILATION_UNIT := ASIS_COMPILATION_UNITS.LIBRARY_UNIT ('SPACE_CRAFT_MANAGER', ...)

  COMPILATION_UNIT now contains a 'with A; with B; package SPACE_CRAFT_MANAGER is ... end SPACE_CRAFT_MANAGER'
- THE_PACKAGE_DECLARATION : = ASIS_COMPILATION_UNITS.UNIT_DECLARATION (COMPILATION_UNIT);

  PACKAGE_DECLARATION now contains 'package SPACE_CRAFT_MANAGER is ... end SPACE_CRAFT_MANAGER'
- DECLARED_ITEMS : = ASIS_DECLARATIONS.VISIBLE_PART_DECLARATIVE_ITEMS (PACKAGE_DECLARATION);

  DECLARED_ITEMS is a list of declarations from the package spec. We now search the list for the identifier 'SPACE_CRAFT_ID_TYPE'. When we have a match, we retrieve from the list the type declaration.
- TYPE_DECLARATION := DECLARED_ITEMS(I);

  TYPE_DECLARATION is now 'type SPACE_CRAFT_ID_TYPE is (SAT1, SAT2, SAT3);'

History of ASIS

Feb 1990	- Telesoft implemented a prototype ASIS
Nov 1990	- STARS contracted with Telesoft to define ASIS
Mar 1991	- Rational released LRM interfaces to STARS
May 1991	- ASIS 0.3 delivered to STARS
Jan 1992	- Telesoft releases ASIS as a product
Jul 1992	- AJPO endorses and sponsors ASIS to become a standard
Nov 1992	- ASIS Working Group chartered under SIGAda to develop an ASIS standard

IBM’s Use of ASIS - Recorded Data Analysis

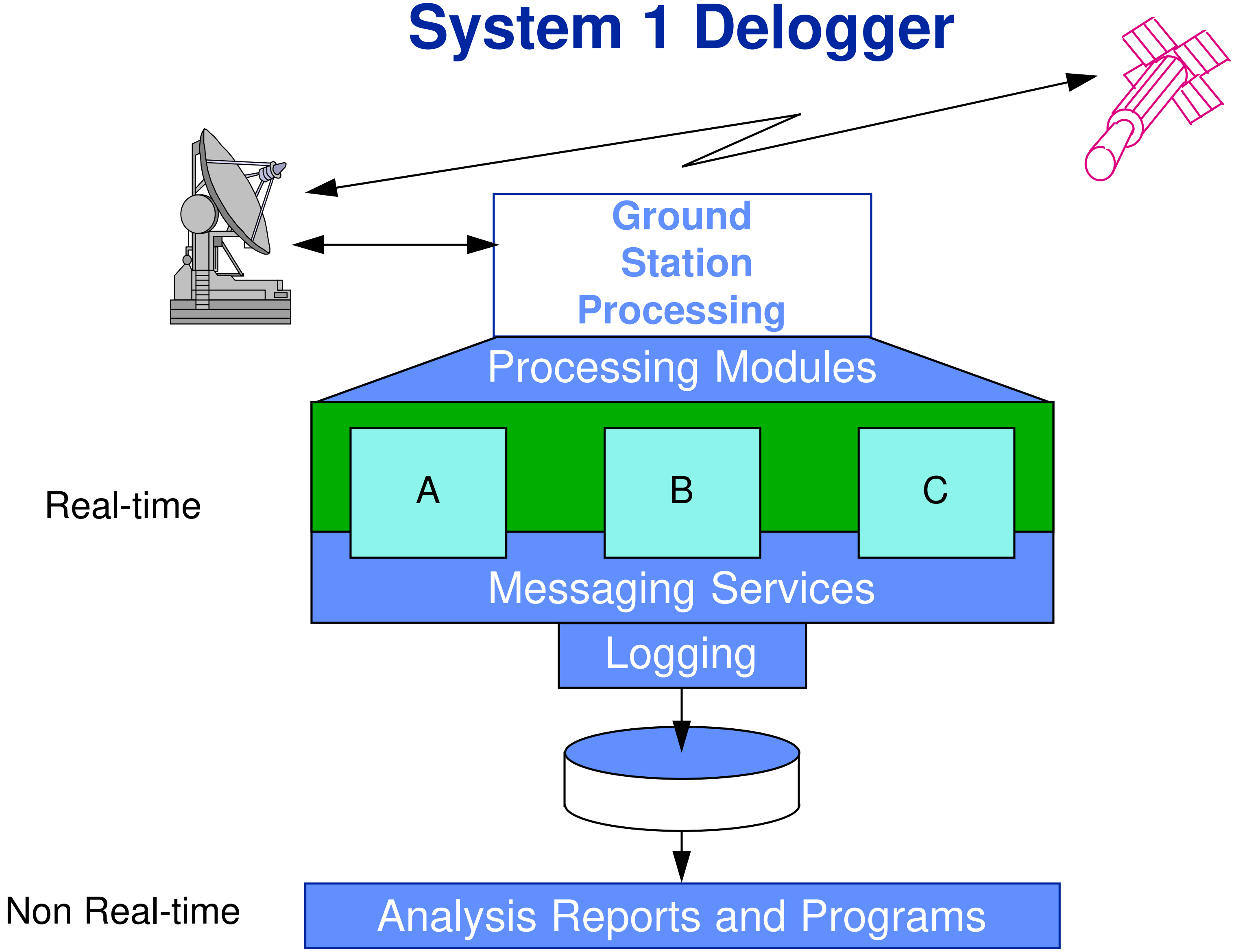
Problem: need to analyze satellite data recorded from the real time system.

Requirement: ONE analysis application to analyze ALL data structures without ‘withing’ the numerous and complex Ada types from the real time system.

Solution: By knowing the name of the Ada record type, use ASIS to query the compiler library and determine the components (and recursively the subcomponents) and physical layout (representation attributes) of the data structure.

- Mar 1992 - evaluated ASIS as a solution
- May 1992 - Telesoft adds functionality (representation attributes)
- May 1992 - Began implementation
- Sep 1992 - Completed production application.

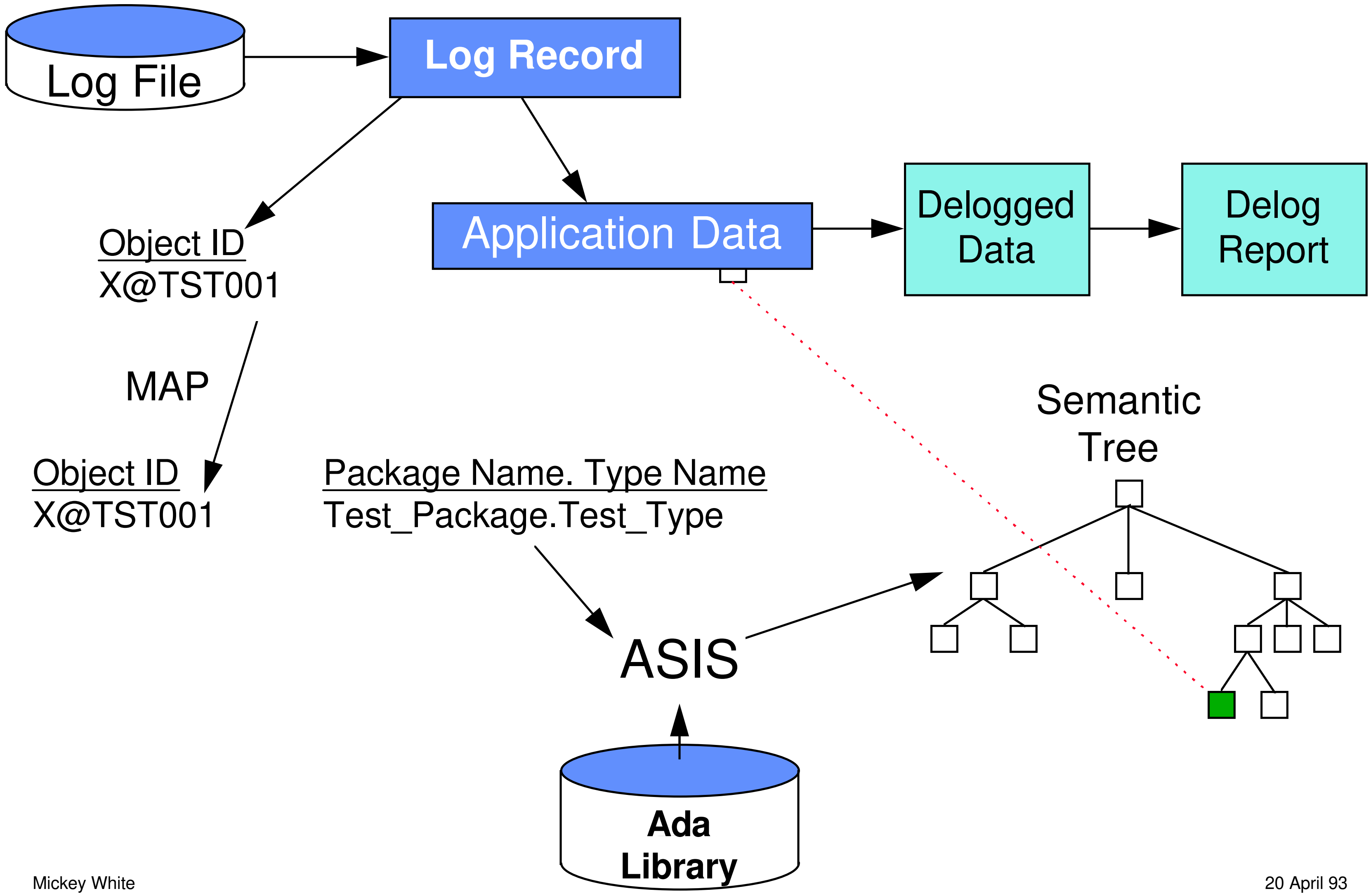
System 1 Delogger



Benefits of ASIS for Recorded Data Analysis

- Allows real time data types to be of arbitrary complexity
  - Handles dynamic types (discriminated records)
  - Handles generic instantiations
  - Handles private types, and subtypes of private types
  - Handles any level of nesting records and arrays
  - Arrays can be constrained by a discriminant, constant, or can be of a static type
  - Handles representation clauses on types
- 704 hours for building a ‘types dictionary’ reduced to 2 hours (each major build had to rebuild the dictionary)
- 95K SLOC source code expansion for types dictionary reduced to 0K SLOC

Universal Delogger



Standardization of ASIS - Status

- ASIS 1.1 to be finalized this month
  - AJPO supports ASIS 1.1 as a defacto standard for Ada 83
  - To be submitted to ISO WG9 as a committee draft in June
- Work on ASIS 2.0 for Ada9X to begin in July
- Vendor implementations - contact your compiler vendor
  - Alsys (Telesoft) - available
  - Alsys - this year
  - Rational - this year
  - Verdix ?
  - DEC ?

ASIS Working Group Information

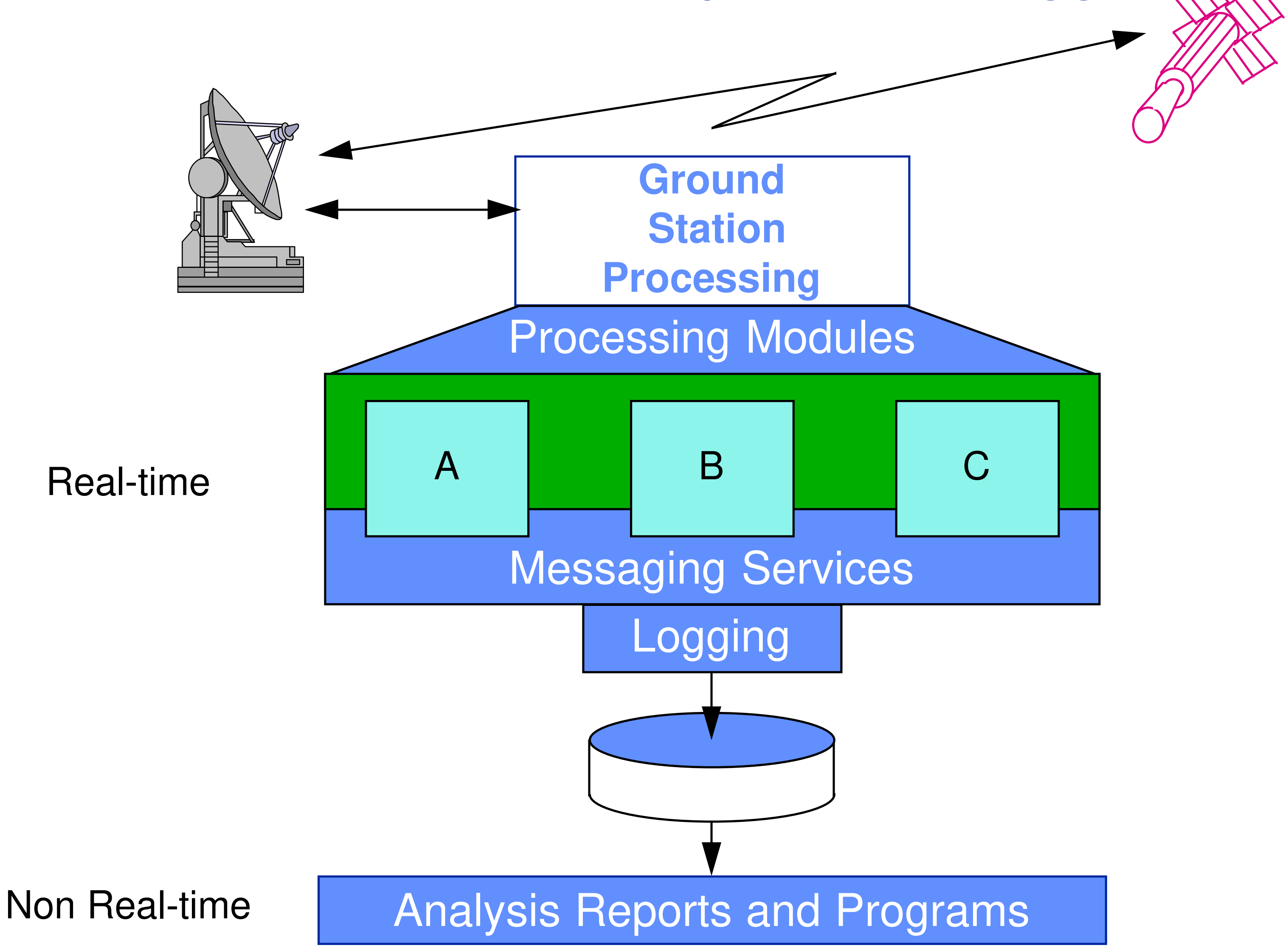
- Open to any interested parties
- Contact Tom Shields (chair): shields@stars.reston.paramax.com to be put on the Internet mailing list*

Next meeting will be in conjunction with WAdaS'93, July 1-2, in Washington D.C.**

* note now to *asis-request@sw-eng.falls-church.va.us*
or to *asis-technical-request@sw-eng.falls-church.va.us*

** For current information, please see the ASIS Home Page at URL:
http://www.acm.org/sigada/WG/asiswg/asiswg.html

Attachment 1: System 1 Delogger



Attachment 2: Messages

Header	Object - Application Data
	Byte_String (Array of Bytes)

Message Structure

- Header
    - Identifies the Ada type name associated with the Application Data
- Application Data
    - Instance of an Ada type, converted to a 'Byte_String'

Attachment 3: Example Application Data Types

```
-- MESSAGE 1
type TEST_TYPE1 (DISCRIM1: INTEGER) is
     private;

--MESSAGE 2
type TEST_TYPE2 is
     record
          FIELD1 : TEST_TYPES_PACKAGE2.RECORD_TYPE;
          FIELD2 : TEST_TYPES_PACKAGE2.ARRAY_TYPE;
          FIELD3 : TEST_TYPES_PACKAGE2.PRIVATE_NEGATIVE_TYPE;
          FIELD4 : TEST_TYPES_PACKAGE2.PRIVATE_ENUM_TYPE;
     end record;

private

     subtype RANGE_TYPE is POSITIVE range 10 .. 20;
     type    ARRAY_TYPE is (INTEGER range <>) of FLOAT;

     type TEST_TYPE1 (DISCRIM1: INTEGER) is
          record
               FIELD1: RANGE_TYPE;
               FIELD2: ARRAY_TYPE (1 .. DISCRIM1);
          end record;
```