API Birds of a Feather
How do we work together
on APIs

- Commercial developers provide sources and at no cost for single and educational use.
- GNU licenses include: Other forms of licensing will be considered.

Windows XP and XP Embedded Device Drivers

- Presently can not use Ada including A# to write to a port.
- Register (Port) Programming ?
  - How does Ada send information to control a board?
- Is a port a new memory address?

package System **is**
type Address **is** *implementation-defined*;

Implementation Advice

Address should be of a private type.

- **Reason:** This promotes uniformity by avoiding having implementation-defined predefined operations for the type. We don't require it, because implementations may want to stick with what they have.

Implementation Note: It is not necessary for Address to be able to point at individual bits within a storage element. **Nor is it necessary for it to be able to point at machine registers.** It is intended as a memory address that matches the hardware's notion of an address.

Solution

- Make Address a generic private type and Instantiate it for:
  - the CPU's main memory
  - Ports
  - Shared memory
  - Networked memory
- This will need protection procedures.

XML Birds of a Feather

- Simplify the bidirectional conversion between XML and Ada by making the data types identical.
- Numbers
- Ada.Strings.Bounded

- Standard's parsimony
  - Create new by reusing old standards.

Numbers

- Create the same numeric types for both.
- Ecumenical approach use ECMA types.
- http://www.ecma.ch/
- Originally, European Computer Manufacturers Association
- Now, ECMA International - European association for standardizing information and communication systems.

Real Types are Primitive

XML Type	Source	ECMA
float	IEEE single-precision 32-bit	float32
double	IEEE double-precision 64-bit	float64
*decimal	W3C	decimal
*Minimum of 18 Digits. Pentium uses double extended precision floating point registers. PowerPC uses floating point registers for fixed-point.		

Integer Types are Derived

Type & Derivation Sequence	ECMA	Min-Inclusive	Max-Inclusive
*integer		-infinity	infinity
nonPositiveInteger			0
negativeInteger			-1
long	Int64	-2**63	(2**63) -1
int	Int32	-2**31	(2**31) -1
short	Int16	-2**15	(2**15) -1
byte	SByte ***Int8***	-2**7	(2**7)-1

*Derived From decimal; fractionDigits·= 0

Integer Types Cont.

Type & Derivation Sequence	ECMA	Min Inclusive	Max Inclusive
*nonNegativeInteger		0	infinity
unsignedLong	UInt64	0	2**64-1
unsignedInt	UInt32	0	2**32-1
unsignedShort	UInt16	0	2**16-1
unsignedByte	Byte ***UInt8***	0	2**8-1
positiveInteger		1	infinity

*Derived from Integer

XML 32 bit Integer Equivalent

```
<simpleType name="Int32_Type">
  <restriction base="int">
    <minInclusive value="-2147483648"/>
    <maxInclusive value="2147483647"/>
  </restriction>
</simpleType>
```

Ada Int32 Type & Subtypes

```
subtype Int32 is Integer;
--or
type Int32 is range -2**31..2**31-1;
   for Int32'SIZE use 32;

subtype Natural_32  is Int32
    range 0..Int32'Last;
subtype Positive_32 is Int32
    range 1..Int32'Last;
```

Create XML Strings by Addition of fields to Bounded_String

- Encapsulated in generic packages, Ada.Strings.Bounded. & Wide_Bounded
- Solution:

1. Create a generic that instantiates Ada.Strings.Bounded with a generic type.
2. Add a Character_Set_Type etc. to a private tagged type.
3. Add a Modified version of all of the methods in Ada.Strings.Bounded

How to Create a Character Set

```
Latin_1_Range  : constant
   Str_Maps.Character_Range :=
        (Low => Latin_1.Null,
        High => Latin_1.Lc_Y_Diaeresis);
Latin_1_Char_Set :   Character_Set_Type  :=
   Str_Maps.To_Set (Span => Latin_1_Range);
```

XML Bounded Strings with Character Sets

```
with Ada.Strings.Bounded;
with Ada.Strings;
with Character_Sets;
with Pattern_Pkg;
generic
   Max_Bd_Length : Positive;
   Character_Set :
        Character_Sets.Character_Set_Type
           := Character_Sets.Latin_1_Char_Set;
```

```
Min_Bd_Length : Positive
  := Min_Bd_Length;  ---1
Pattern      :  Pattern_Bd_Type    :=
        Null_Pattern_Bd;
```

Generic Instantiation

```
package Generic_Bd_W_Char_Sets is
……….
package Generic_Bd_Strings is new
Ada.Strings.Bounded.Generic_Bounded_Length
   (Max => Max_Bd_Length);

subtype Generic_Bd_Type is
   Generic_Bd_Strings.Bounded_String;
```

```
private
  type Generic_Bd_W_Char_Set_Type is tagged
    record
      Generic_Bd_Part    : Generic_Bd_Type    :=
        Null_Generic_Bd;
      Character_Set_Part : Character_Set_Type :=
        Character_Set;
      --This permits the Character_Set to be
      --specified at instantiation and defaults to
      --Latin_1
      Min_Bd_Length_Part : Positive          := 1;
      Pattern_Part      : Pattern_Bd_Type    :=
            Null_Pattern_Bd;
    end record;
```

Problem

- XML is based on Unicode
  - UTF-8,
  - UTF-16,
  - UTF-32

Solution

Ada.Strings.Unbounded, Ada.Strings.Bounded Ada.Strings.Maps & Ada.Characters.Handling need to have added 32 bit versions.

Translation between 8, 16 and 32 bit types

Briot's XML/Ada probably can be made to work with To_String (Img) of bounded strings.

What is new in XML?

- Office 2003
  - Word & Excel can work in XML mode.
  - Based on Schema
  - Uses XSL (Extensible Stylesheet Language) for transformations.
  - Does NOT use XSL fo (Formatting Objects)
  - Does NOT use XML SVG (Scalable Vector graphics)
  - The formatting is together with the XML
    - Can be convoluted!
    - Microsoft's extensive use of abbreviations results in extensive use of documentation (comments).

XForms 1.0
W3C Recommendation 14 October 2003

- Xforms: model, instance data, and user interface
- Separates presentation from content
- Benefits: reuse, strong typing, reduction of round-trips to the server, device independence, and a reduces the need for scripting.
- Xforms
  - Not a free-standing document type
  - Integrated into XHTML or SVG.
- Examples from http://www.formsplayer.com/

XML Web packages

- Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language (Working Draft 10 November 2003
- Provides a model and an XML format for describing Web services.
- Separate the description of the abstract functionality offered by a service from concrete details of a service description such as "how" and "where" that functionality is offered.

- WSDL 2.0 Message Exchange Patterns define the sequence and cardinality of abstract messages sent or received by an operation.
- The WSDL Version 2.0 Part 3: defines a language for describing such concrete details for SOAP 1.2
- Ada Distributed Systems Annex or CORBA replace IDL with XML?

Odds & Ends

- I Translated Thomas Wolf's AdaBrowse Document Type Definition (DTD) into a Schema.
- A complete description of Ada in XML schema would permit the use of an XML based word processor as a program editor.
  - The design documentation and the source could be connected by hypertext links.