Avatox XML Format[1]

Version 1.8, 26 August 2007

Overview

The Avatox XML Format (AXF from now on) consists predominantly of the ASIS-oriented structure possessed by Ada compilation units. This is not to say that AXF cannot be used to represent other programming languages' structures, but it is geared towards Ada.

The compilation unit is wrapped in a <codeRepresentation> element that also includes supporting information about the contents of the representation.

Each XML element representing a software construct has associated with it a “pedigree” indicating its membership in one of various classes of representational elements, such as whether the element is derived from a standard, or from a vendor's value-added additions. The purpose of the pedigree specification is to allow AXF processing tools to filter out content they don't know how to deal with, and allows producers of AXF to add enhanced content to the AXF for their own use, without diminishing the value of the AXF to others.

Supplemental information to aid AXF transformation is provided via optionally generated "axfPoint" elements--AXF Points Of INformation for Transformation.

Style

The non-ASIS elements of an AXF document utilize the canonical XML "camelback" capitalization and naming style, without underscores. This means that a single-word element or attribute name is all lowercase, while in a multi-word name the first word is lowercase, and the first letter of each subsequent word is capitalized. E.g., "sourceLanguage", "compiler".

The default style for ASIS elements is all capitalized, and corresponds to the 'Image attribute representation for the ASIS enumeration types, e.g. "A_CLAUSE", "AN_EXPRESSION". Avatox provides a "-x" switch that when selected converts this Image representation to XML camelback ("aClause", "anExpression") to maintain visual consistency with the remainder of the AXF document and the typical XML style. Also, when Avatox has been directed to generate axfPoint elements, the letter-casing style is set to camelback to ensure a consistent style throughout the document.

1The Avatox XML format for source code representation is an evolving format, though with version 1.1 the outermost structure was solidified. There is as yet no XML schema, due in no small part to the amount of effort it would require to construct a schema capturing the ASIS-based structure. It is certainly doable, and likely will be created at some point in the future.

AXF Structure

Element: `<codeRepresentation>`

The outermost container element of an AXF document. A codeRepresentation element contains three classes of child elements: sourceElement, pedigrees, and the element that corresponds to a compilation unit (i.e., A_PACKAGE_BODY, A_PACKAGE, etc., or, with XML style capitalization, aPackageBody, aPackage, etc.)

Element: `<sourceLanguage>`

Information about the source language(s) that is represented in the AXF document.

Attributes:

name	The common name of the language, e.g., Ada, C++
languageVersion	Indicate which version or standard of the language, e.g., "95" for "Ada 95", "99" for "C99"
compiler	The compiler for the code. This is free format and may be defined by the compiler vendor. E.g., "GNAT GPL 2006 (20060522-34)"

Element: `<pedigrees>`

The set of one or more pedigree elements that provide characterizing information about the representational elements that make up the code representation.

Element: `<pedigree>`

Characterization information about the elements providing the code representation. A pedigree element can have one of three child element kinds that provide information: standardInfo, processorInfo, manualInfo.

Attributes:

name	The identifier by which the pedigree is referenced.
xmlStyle	Boolean value indicating whether elements of this pedigree will conform to the XML camelback naming style, or use some other, such as the enumeration 'Image representation.

Element: `<standardInfo>`

Child element of <pedigree>. Elements having this pedigree correspond to that of a public standard, and therefore should be processable by any tool that supports that standard. The Ada Semantic Interface

Specification (ASIS) is such a standard.

Attributes:

name	The name of the standard.
implementor	The implementor of that standard with whose aid the elements in the AXF document were acquired.
implementorVersion	The version of the standard implementor's implementation.

Element: `<processorInfo>`

Child element of <pedigree>. Elements having this pedigree were generated by a specific code processor, and may or may not be processable by other tools. For example, Avatox inserts ASIS-like A_COMMENT (or aComment) elements within the ASIS structure to contain comment lines, since ASIS does not explicitly support comment processing.

Attributes:

name	Name of the processor that produces the non-standard content.
implementor	Vendor, or analogous, information about the processor.
implementorVersion	The version of the processor implementation.

Element: `<manualInfo>`

Child element of <pedigree>. Elements having this pedigree were manually inserted by a text editor, or by some other ad hoc mechanism. (It is not unreasonable to expect that manual pedigreed elements may eventually evolve into processor pedigreed elements. Simply altering the pedigree content will accommodate this.)

Attributes:

name	Name, or purpose, of the manual effort.
implementor	Name of the person, group, organization, etc. that performed the manual entry.
implementorVersion	Any sort of versioning, or date stamping, that may be worth recording.

ASIS Structural Elements

The sibling elements that follow the <pedigrees> element is any one of those corresponding to a

compilation unit, and whose content is defined by the standard, such as ASIS, that is representing the code.

Each element within a compilation unit element, including the compilation unit element itself, has the following attributes:

Attributes:

pedigree	Name of the pedigree associated with that element's presence.
startLine	The line of compilation unit text on which that element's span began.
startCol	Column on the startLine that marked the beginning of that element's span.
endLine	Line of compilation unit text on which the final portion of the element's span resides.
endCol	Column on the endLine that marked the ending of that element's span.

Contingent ASIS Annotations

Certain elements representing program structure have additional attributes specific to their kind and position within the program representation. Attributes that are typed as Boolean are generated by Avatox only when their value is “true”. While absence implies a value of false, stylesheets should be constructed to not simply assume that the presence of the attribute indicates it has a true value.

Attributes:

accessibility — The accessibility of a given declaration to internal and external entities. Accessibility can take on one of four values:

- “universal” - The entity is accessible throughout the subsystem namespace.
- “packaged” - The entity resides in the outermost packaging construct. For Ada, C++, and Java this would be the public parts of their package or class.
- “internal” - The entity is accessible within its enclosing unit and those descended from it. For Ada this corresponds to the “private” part, for C++ and Java this would be the “protected” section.
- “private” - The entity is accessible only within its enclosing unit. For Ada this would be within a unit body, for C++ and Java it would be those declarations in the class' private part.

isAStatement — Boolean attribute designating whether an entity that is **not** a statement should be considered as one. In Ada this applies only to pragmas, e.g.,

	Assert.
prefixNotation	Boolean attribute indicating whether the *object.method* syntax is being used for a subprogram call.

AXF Supplemental Information

The generation of axfPoint elements is triggered by specifying one ore more of the Avatox -axf*xxx* switches, or the all-selecting -a switch. axfPoint ("AXF Point Of INformation for Transformation") elements are context-dependent elements that provide supplemental information pertaining to the element in which they're nested.

Element: `<axfPoint>`

Context-sensitive element whose content applies to its enclosing element.

Attributes:

pedigree	Name of the pedigree associated with the axfPoint element.
axfNumber	Programming language independent representation of numeric literals, both integer and float. Ada-style based literal notation is used for non base-10 numbers, as it can represent any numeric base from 2 to 16, rather than just 2, 8, 10, and 16.
axfOper	Enumeration of operator symbol names. See the file vatox-axf_points-terminal-nomenclature.ads for the current list.
axfScope	A blank separated list of the identifier and its enclosing context names. The primary purpose is to provide a unique ID for use in distinguishing amongst identifiers having the same name. Nested within each axfScope axfPoint element are additional axfPoint elements, one for each level of nesting. These subordinate axfPoint element attributes are:
	axfScopeName — The name of each enclosing scope, with the lowermost one being the identifier itself.
	axfScopeLevel — The scoping level at which the scope resides, with 0 being the outermost scope of the compilation unit.
axfXRef	A cross-referencing name consisting of a blank separated list of the identifier and its enclosing context names that correspond to the axfScope of the referenced identifier.

axfXRefName	The name of each enclosing scope, except for the lowermost one, which is the identifier itself.
axfXRefLevel	The scoping level at which the scope resides, with 0 being the outermost scope of the compilation unit.

Example

The following is an example, from vatox.adb.axf, of the opening content of the AXF file up through the first "with" clause (with comments omitted):

```
<?xml version="1.0" encoding="UTF-8" ?>
<codeRepresentation>
   <sourceLanguage name="Ada" languageVersion="95" compiler="GNAT GPL 2006 (20060522-34)"/>
   <pedigrees>
      <pedigree name="axfPoint" xmlStyle="true">
         <processorInfo name="Avatox" implementor="McKae Technologies (www.mckae.com)"
                                                                  implementorVersion="1.6"/>
      </pedigree>
      <pedigree name="asis" xmlStyle="true">
         <standardInfo name="Ada Semantic Interface Specification (ASIS)"
                        implementor="AdaCore (http://www.adacore.com)"
                        implementorVersion="ASIS 2.0.R for GNAT GPL 2006 (20060522-34)"/>
      </pedigree>
      <pedigree name="avatoxAsis" xmlStyle="true">
         <processorInfo name="Avatox"
                        implementor="McKae Technologies (www.mckae.com)"
                        implementorVersion="1.6"/>
      </pedigree>
   </pedigrees>
   <aPackageBody pedigree="asis" unitName="Vatox.Traversal" startLine="29" startCol="1"
                                                                  endLine="879" endCol="20">
      <aClause pedigree="asis" startLine="29" startCol="1" endLine="29" endCol="32">
         <aWithClause pedigree="asis" startLine="29" startCol="1" endLine="29" endCol="32"/>
         <anExpression pedigree="asis" startLine="29" startCol="6" endLine="29" endCol="31">
            <aSelectedComponent pedigree="asis" startLine="29" startCol="6" endLine="29" endCol="31"/>
            <anExpression pedigree="asis" startLine="29" startCol="6" endLine="29" endCol="19">
               <aSelectedComponent pedigree="asis" startLine="29" startCol="6"
                                                        endLine="29" endCol="19"/>
               <anExpression pedigree="asis" startLine="29" startCol="6" endLine="29" endCol="8">
                  <anIdentifier pedigree="asis" ident="Ada" startLine="29" startCol="6"
                                                        endLine="29" endCol="8"/>
                  <axfPoint pedigree="axfPoints" axfXref="Ada">
                     <axfPoint pedigree="axfPoints" axfXrefName="Ada" axfXrefLevel="0"/>
                  </axfPoint>
               </anExpression>
               <anExpression pedigree="asis" startLine="29" startCol="10" endLine="29" endCol="19">
                  <anIdentifier pedigree="asis" ident="Characters" startLine="29" startCol="10"
                                                        endLine="29" endCol="19"/>
                  <axfPoint pedigree="axfPoints" axfXref="Ada.Characters">
                     <axfPoint pedigree="axfPoints" axfXrefName="Ada.Characters" axfXrefLevel="0"/>
                  </axfPoint>
               </anExpression>
            </anExpression>
            <anExpression pedigree="asis" startLine="29" startCol="21" endLine="29" endCol="31">
               <anIdentifier pedigree="asis" ident="Conversions" startLine="29" startCol="21"
                                                        endLine="29" endCol="31"/>
               <axfPoint pedigree="axfPoints" axfXref="Ada.Characters.Conversions">
                  <axfPoint pedigree="axfPoints"
                        axfXrefName="Ada.Characters.Conversions" axfXrefLevel="0"/>
               </axfPoint>
            </anExpression>
         </anExpression>
      </aClause>
```

Contributions and suggestions for improvement and enhancement are gladly accepted and will be considered for incorporation into future revisions of the AXF format.

Marc A. Criley
Mckae Technologies

18 April 2007