Abbreviated syntax In XPATH, shortened SYNTAX for a LOCATION STEP. See Figure A.1. *Contrast* UNABBREVIATED SYNTAX.

Absolute expression *See* ABSOLUTE LOCATION PATH.

Abbreviated syntax	Unabbreviated syntax
nothing	`child::`
`@`	`attribute::`
`//`	`/descendant-or-self::node()/`
`.`	`self::node()`
`..`	`parent::node()`

Figure A.1 Some examples of abbreviated syntax.

Absolute location path In XPATH, a LOCATION PATH that starts at the ROOT NODE of a NODE TREE. An absolute location path expression begins with a forward slash (which indicates the root node) and is followed by a RELATIVE LOCATION PATH. An absolute location path is a sequence of LOCATION STEPS, each separated by a forward slash. In the example shown in Figure A.2 the absolute location path selects all the "price" elements of all the "book" elements of the "catalog" element. Note that the "catalog" ELEMENT is the ROOT ELEMENT of the XML DOCUMENT and should not be confused with the ROOT NODE of the XPATH NODE TREE represented by the start forward slash. *Contrast* RELATIVE LOCATION PATH.

Action attribute *See* XFORMS MODEL.
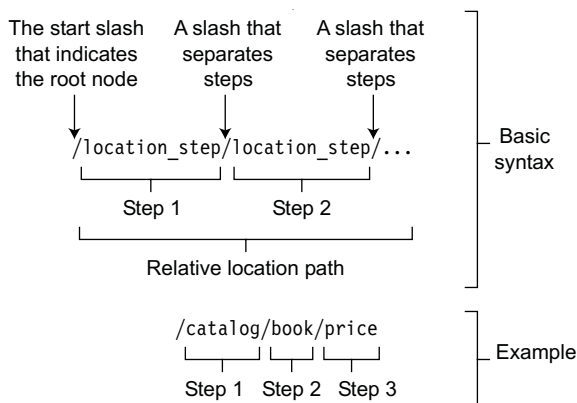
Active intermediary *See* SOAP INTERMEDIARY.

Figure A.2 Absolute location path.

```
         <catalog xmlns:xlink="http://www.w3.org/1999/xlink"
 XLink ┌ xlink:type="simple"
 type └ xlink:href="Catalog.xml" ┐— Ending resource
  The   xlink:show="new"
"actuate"┌ xlink:actuate="onRequest"/>
attribute
```

Figure A.3 An example of an XLink "actuate" attribute.

Actuate attribute In XLINK, an ATTRIBUTE used to specify when and how the LINK should be activated. Its main ATTRIBUTE VALUES are: (1) `onRequest` – some user's action is required to activate the link, such as a mouse click (see Figure A.3); (2) `onLoad` – the link is activated automatically immediately on loading the STARTING RESOURCE.

ADML (Architecture Description Markup Language) An XML-BASED MARKUP LANGUAGE for the interchange of architectural descriptions between a variety of architectural design tools. More details about ADML are available at `http://www.opengroup.org/architecture/adml/adml_home.htm`.

aecXML (Architecture, Engineering and Construction XML) An XML-BASED MARKUP LANGUAGE for representing information in the Architecture, Engineering and Construction (AEC) industry. Details of aecXML can be found at `http://www.iai-na.org/aecxml/mission.php`.

Agent *See* INTELLIGENT AGENT.

Agent-Oriented Rule Markup Language *See* AORML.

Aggregation & Logging of User Requests for assistance Extensible Markup Language *See* ALUREXML.

AIML (Astronomical Instrument Markup Language) An XML-BASED MARKUP LANGUAGE for describing and controlling astronomical instruments. AIML is a domain-specific implementation of IML. More details about AIML can be obtained from `http://pioneer.gsfc.nasa.gov/public/aiml/`.

All Different statement In OWL, a statement that indicates all given INDIVIDUALS are different from each other. In the example shown in Figure A.4, "Mike", "Chris" and "Dan" are stated to be all different individuals of the CLASS "Person". *See also* SAME AS STATEMENT; DIFFERENT FROM STATEMENT.

all element In an XML SCHEMA, a special ELEMENT that is used as an INDICATOR that specifies by default that the CHILD ELEMENTS of the declared element may appear in any order, and also that each child element must occur once and only once. The QUALIFIED NAME of the "all" element can be either "xs:all" or "xsd:all", depending whether the XS NAMESPACE PREFIX or XSD NAMESPACE PREFIX is being used. See Figure A.5.
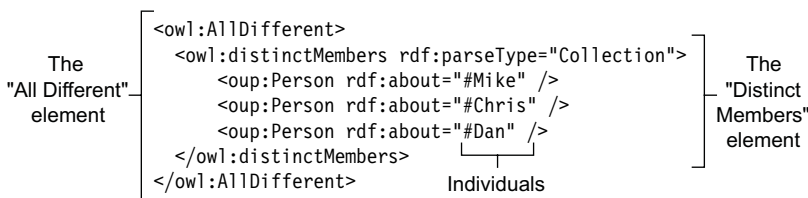
```
                    ┌ <owl:AllDifferent>
  The               │   <owl:distinctMembers rdf:parseType="Collection">        ┐  The
"All Different"      │     <oup:Person rdf:about="#Mike" />                      │ "Distinct
 element             │     <oup:Person rdf:about="#Chris" />                     │ Members"
                    │     <oup:Person rdf:about="#Dan" />                       ┘  element
                    │   </owl:distinctMembers>           Individuals
                    └ </owl:AllDifferent>
```
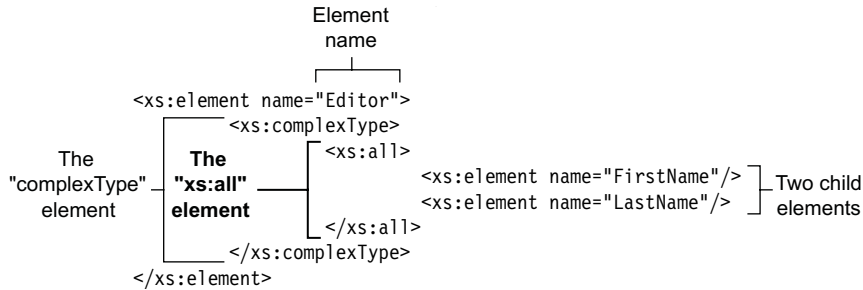
Figure A.4 Stating that individuals are all different.

```
                            Element
                             name
                            ┌───┴───┐
            <xs:element name="Editor">
              ┌─── <xs:complexType>
  The         │    ┌── <xs:all>
"complexType"─┤  The │        <xs:element name="FirstName"/> ┐ Two child
  element     │ "xs:all"─┤    <xs:element name="LastName"/>  ┘ elements
              │ element  │
              │    └── </xs:all>
              └─── </xs:complexType>
            </xs:element>
```

Figure A.5 The use of the "all" element.

all indicator *See* INDICATOR.

Alt (Alternative) *See* ALTERNATIVE CONTAINER.

Alt class In RDF SCHEMA, the rdf:Alt CLASS is the class of ALTERNATIVE CONTAINERS. It is a SUBCLASS of the CONTAINER CLASS.

Alternative container In RDF, the rdf:Alt CONTAINER is a container that is a collection of RESOURCES or LITERALS that represent alternative values, such as alternative language translations for the title of a book. See Figure A.6. Note that in the rdf:li ELEMENT, the "li" LOCAL PART stands for "a list item".

Alternative element *See* ALTERNATIVE CONTAINER.

ALUReXML (Aggregation & Logging of User Requests for assistance Extensible Markup Language) An XML-BASED MARKUP LANGUAGE that allows any Web-based "user assistance" product or system to log information on specific problems that customers have. For more details, see http://www.alurexml.org/alurexml.htm.

AML (Avatar Markup Language) An XML-BASED MARKUP LANGUAGE for describing avatar-based communication, including facial and body animation as well as text-to-speech content. (An avatar is the visual character you use to represent yourself in Virtual Reality). More details about AML are available at http://ligwww.epfl.ch/~aguye/AML/.

AnatML (Anatomical Markup Language) An XML-BASED MARKUP LANGUAGE for describing anatomy, especially for storing geometric information about the human musculoskeletal system. Details of AnatML are available at http://www.physiome.org.nz/sites/physiome/anatml/pages/.

Anatomical Markup Language *See* ANATML.

Ancestor In XPATH, an ancestor of the CONTEXT NODE, such as the PARENT of context node, the parent's parent, and so on. *See also* ANCESTOR AXIS.

ancestor axis In XPATH, an AXIS that selects the PARENT of the CONTEXT NODE, the parent's parent, and so on. The ancestor axis always includes the

```
            <rdf:Description about="http://www.springer.de/spc/book1">
               <dc:Title>
             ┌── <rdf:Alt>
The "rdf:Alt"│      <rdf:li xml:lang="en"> English Title</rdf:li>
  element ───┤      <rdf:li xml:lang="de"> Deutscher Titel</rdf:li>
             └── </rdf:Alt>
               </dc:Title>
            </rdf:Description>
```
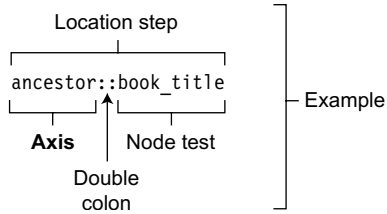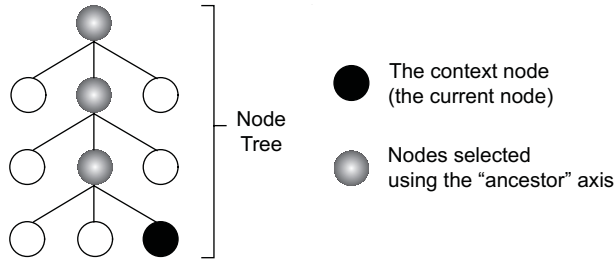
Figure A.6 An example of an RDF "Alternative" container.
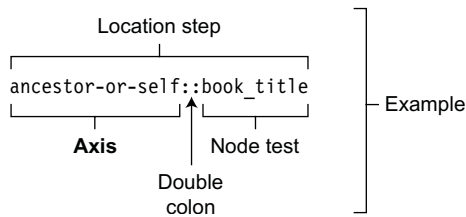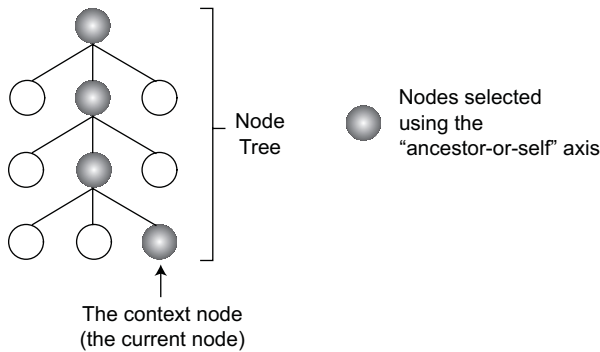
Figure A.7 The "ancestor" axis.

Figure A.8 The "ancestor-or-self" axis.

ROOT NODE, unless the context node is the root node. See Figure A.7.

ancestor-or-self axis In XPATH, an AXIS that selects the same NODES as the ANCESTOR AXIS and also the CONTEXT NODE itself. See Figure A.8.

Angle brackets Special signs used for delimiting a MARKUP TAG and differentiating it from the content
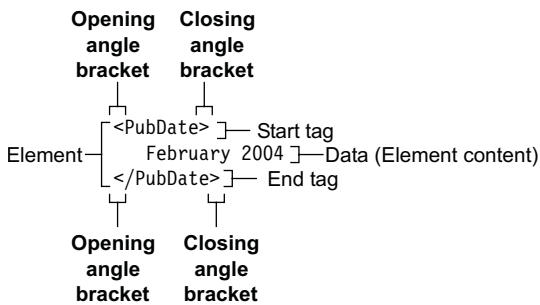
6

Figure A.9 Opening and closing angle brackets in XML.

of an XML, HTML or SGML document, as shown in Figure A.9. In this respect, angle brackets are a symbol of the so-called "XML revolution". The opening angle bracket (<) and the closing angle bracket (>) are also often called the "less-than" sign (<) and the "greater-than" (>) sign respectively. Because the signs (<) and (>) are used in XML MARKUP not for comparing numbers and values, but as paired DELIMITERS, the use of the term "angle brackets" is probably preferable. See Figure A.9.

Annotated XML The specification of XML, annotated by Tim Bray. More information is available at http://www.xml.com/axml/axml.html.

annotation element A special ELEMENT of the XML SCHEMA language that is intended for commenting XML schemas and also for adding information for applications similar to PROCESSING INSTRUCTIONS. To this end, it has two children, as shown in Figure A.10: (1) the DOCUMENTATION ELEMENT and (2) the APPINFO ELEMENT.
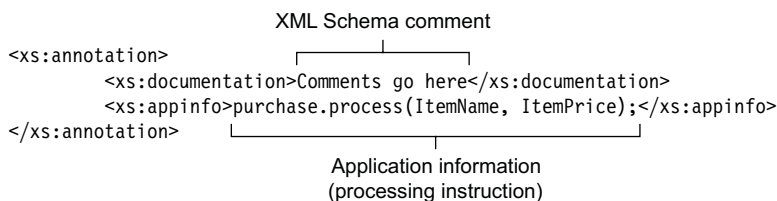
ANY keyword In a DTD, a keyword used in the ELEMENT TYPE DECLARATION to specify that elements of this type may contain all kinds of MARKUP and XML DATA, in any mixture.

Anonymous data type In an XML SCHEMA, a DATA TYPE that has no name and therefore cannot be explicitly referenced. *Contrast* NAMED DATA TYPE.

AORML (Agent-Oriented Rule Markup Language) An XML-BASED MARKUP LANGUAGE for describing agent-oriented business rules in the context of Agent Object Relationship (AOR) models. More details of AORML may be obtained from http://tmitwww.tm.tue.nl/staff/gwagner/AORML/. *See also* RULEML.

Apache Software Foundation *See* ASF.

API (Application Programming Interface) A standardized set of functions and commands that allows any developer to interface an APPLICATION program with other applications.

appinfo element A special ELEMENT of the XML SCHEMA language, used for inserting instructions for applications similar to PROCESSING INSTRUCTIONS into an XML SCHEMA. This element must be NESTED within of the ANNOTATION ELEMENT, as shown in Figure A.11. The "appinfo" abbreviation stands for "application information".

Application An application (also called an application program) is a computer software program that allows the user either to perform useful work not related to the computer itself (for example, a
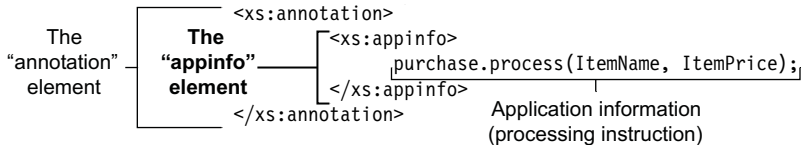

Figure A.10 The structure and use of the "annotation" element.

Figure A.11 The "appinfo" element.

word processor or an XML EDITOR), or to develop other software. This differs from the system software that is used by the computer for its own purposes. Any XML-BASED MARKUP LANGUAGE is officially called an XML APPLICATION, though such terminology is perhaps somewhat misleading.

Application information *See* APPINFO ELEMENT.

Application program *See* APPLICATION.

Application Programming Interface *See* API.

Application services *See* WEB SERVICES.

Application software *See* APPLICATION.

Arc In XLINK, navigable connections between LOCATORS participating in an EXTENDED LINK. It defines the direction of TRAVERSAL and optionally application behavior. There are three types of arc: (1) OUTBOUND ARC; (2) INBOUND ARC; and (3) THIRD-PARTY ARC. Figure A.12 shows an arc between two RESOURCES specified by their locator. This arc connects a particular book to the biography of its author. *See also* ARC-TYPE ELEMENT.
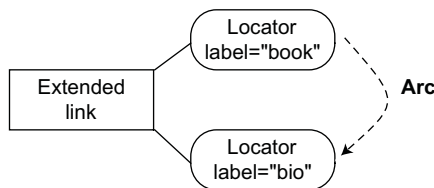
XML/XLink code
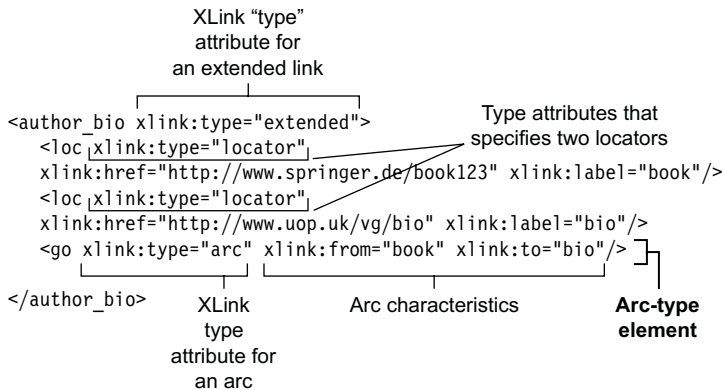
Graphical representation

Figure A.12 An example of an XLink arc.

Archaeological Markup Language *See* ARCHAEOML.

ArchaeoML (Archaeological Markup Language)
An XML-BASED MARKUP LANGUAGE for describing archaeological and geographical data as well as the epigraphic and linguistic features of ancient texts. More details of ArchaeoML may be obtained from `http://www.oi.uchicago.edu/OI/PROJ/ XSTAR/ArchaeoML.html`.

Architecture Description Markup Language
See ADML.

Architecture, Engineering and Construction XML *See* AECXML.

Arc-type element In XLINK, the ELEMENT of an XML DOCUMENT that has a TYPE ATTRIBUTE with the "arc" ATTRIBUTE VALUE and therefore can be used to specify traversal rules among the link's PARTICIPATING RESOURCES. For more details, see ARC; TYPE ATTRIBUTE.

Article Extensible Markup Language *See* AXML.

ASF (The Apache Software Foundation) A not-for-profit corporation that hosts OPEN SOURCE software projects, including the Apache XML project. For more details, see `http://www.apache.org/`.

Associating style sheets with XML documents
A W3C RECOMMENDATION that describes the mechanism of attaching of a STYLE SHEET to an XML DOCUMENT using a special XML style sheet PROCESSING INSTRUCTION. In an XML DOCUMENT, this processing instruction usually follows the XML DECLARATION. The ATTRIBUTE VALUES of its ATTRIBUTES are used to specify whether the attaching style sheet is an XSL STYLE SHEET or a CSS STYLE SHEET, as shown in Figure A.13. This RECOMMENDATION is available at `http:// www.w3.org/TR/xml-stylesheet/`.

Astronomical Instrument Markup Language
See AIML.

Atomic data type In an XML SCHEMA, a DATA TYPE that is considered to be the type that cannot be subdivided into other data types.

ATTLIST declaration (Attribute-list declaration)
A DTD DECLARATION used for specifying the ATTRIBUTES of an ELEMENT. It defines: (1) The number of attributes; (2) ATTRIBUTE NAMES; (3) attribute types, such as CDATA or ENTITY; (4) attribute DEFAULT VALUES, such as "REQUIRED" or "IMPLIED". See Figure A.14 for an example.

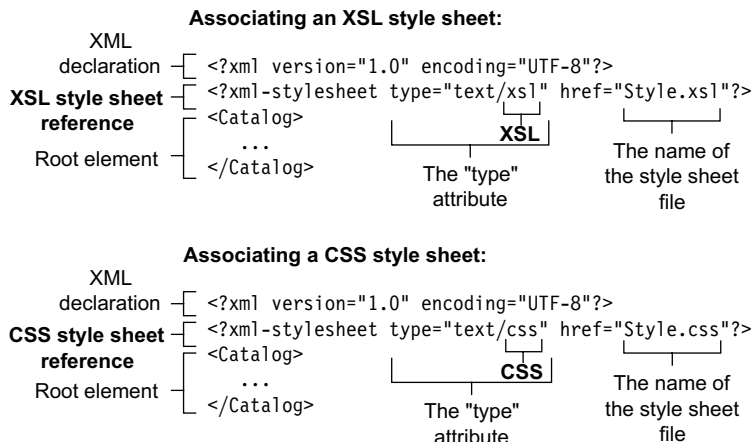**ATTLIST keyword** *See* ATTLIST DECLARATION; DTD DECLARATION.

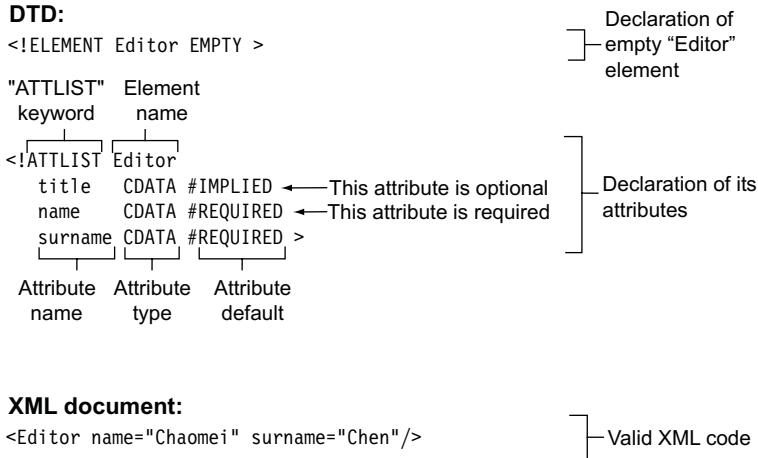**Figure A.13 Examples of associating style sheets with XML documents.**

DTD:

```
<!ELEMENT Editor EMPTY >
```
Declaration of empty "Editor" element

"ATTLIST" keyword   Element name

```
<!ATTLIST Editor
    title   CDATA #IMPLIED    ←—— This attribute is optional
    name    CDATA #REQUIRED   ←—— This attribute is required
    surname CDATA #REQUIRED >
```
Declaration of its attributes

Attribute name   Attribute type   Attribute default

XML document:

```
<Editor name="Chaomei" surname="Chen"/>
```
Valid XML code

Figure A.14 An example of an ATTLIST declaration.

Attribute (name–value pair)

```
<Price currency="USD">79.95</Price>
```
Attribute name   Attribute value

Quotation marks

```
<Price currency="USD">79.95</Price>
```
Equals sign
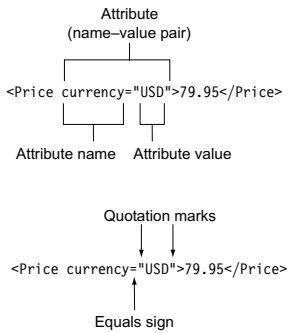
Figure A.15 The anatomy of an attribute.

Attribute A STRUCTURAL CONSTRUCT of XML that consists of a NAME–VALUE PAIR enclosed within the START TAG of an ELEMENT, as shown in Figure A.15. As a property of an element, an attribute provides additional information about the element and modifies certain features of it. Accordingly, an attribute is not as autonomous as an element and makes sense only in the context of the element it belongs to. Attributes can contain any CHARACTER DATA, including WHITE SPACE. Unlike in HTML, ATTRIBUTE VALUES in XML must be surrounded by either double or single quotes. An element can have any number of attributes providing all of them have a unique attribute name. *See also* ATTRIBUTE NAME; ATTRIBUTE VALUE; ATTRIBUTE TYPE; ELEMENT; NAME–VALUE PAIR; START TAG.

attribute axis In XPATH, an AXIS that selects the ATTRIBUTE NODES of the CONTEXT NODE. See Figure A.16.

Attribute declaration An XML SCHEMA COMPONENT used to declare an ATTRIBUTE of an XML DOCUMENT element. All attributes are SIMPLE TYPES, since their ATTRIBUTE VALUES can contain only CHARACTER DATA. On the other hand, each ELEMENT that has one or more attributes is a COMPLEX DATA TYPE. An attribute declaration always appears as the very last part of a COMPLEX TYPE DECLARATION after everything else has been declared. Figure A.17 shows the ELEMENT DECLARATION of the PlanState element, which is an EMPTY ELEMENT. It is of a complex data type and has one attribute named "agreed". *See also* XML SCHEMA COMPONENT; BOOLEAN DATA TYPE.

Attribute declaration *See* ATTLIST DECLARATION.

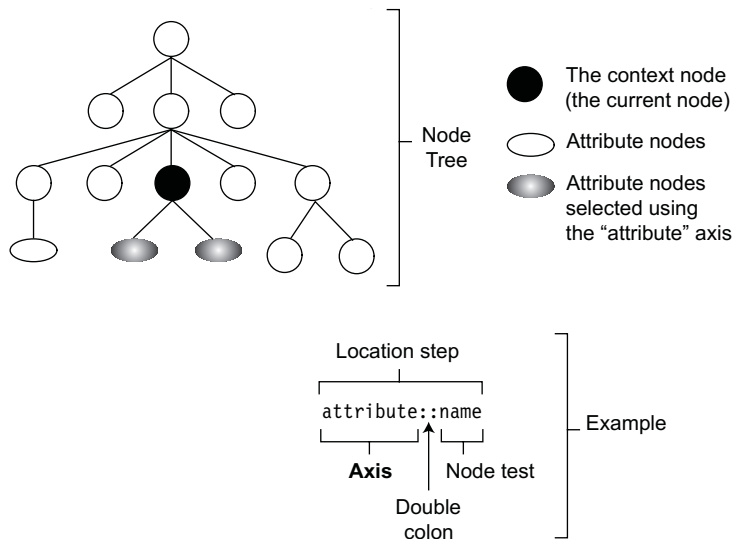**Attribute default value** *See* DEFAULT VALUE.
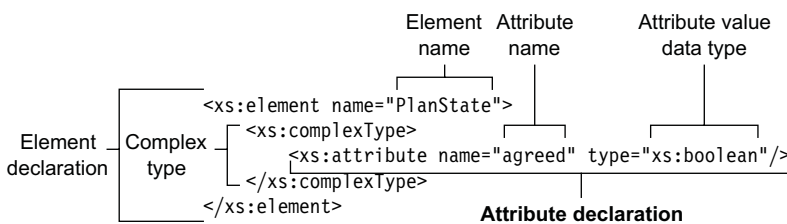
Figure A.16 The "attribute" axis.

Figure A.17 An example of an attribute declaration.

attribute element The "attribute" element is a special ELEMENT of the XML SCHEMA used in ATTRIBUTE DECLARATION for declaring ATTRIBUTES.

Attribute information item One of the eleven types of INFORMATION ITEMS in the INFORMATION SET. Like all of them, it is an abstract description of a STRUCTURAL CONSTRUCT of an XML DOCUMENT. This description is intended for use in other XML-related specifications, which need to conform to the information set. There is an attribute information item in the information set for each ATTRIBUTE of the document. The attribute information item has eight properties, as shown in Figure A.18. For more details, see http://www.w3.org/TR/xml-infoset/.

Attribute name The identifier of an ATTRIBUTE. See Figure A.19. An attribute name is the first part of a NAME–VALUE PAIR. An attribute name must be a VALID XML NAME. This, in particular, means: (1) Attribute names must begin with a letter (but not a number), an underscore (_) or a colon (:); (2) They may not begin with the letters x, m or l in any combination of upper and lower cases, since these are reserved by the W3C for a special use; (3) The use of names that start with a colon (such as :BookTitle) should be avoided, since colons are used only to separate the NAMESPACE PREFIX of a QUALIFIED NAME from its LOCAL PART. *See also* ATTRIBUTE VALUE.
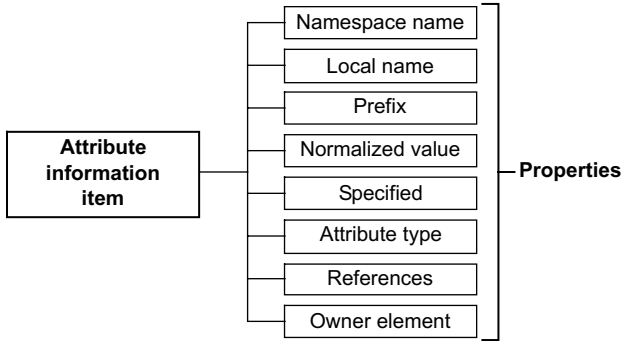
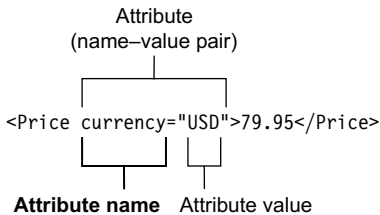**Figure A.18 The attribute information item and its properties.**

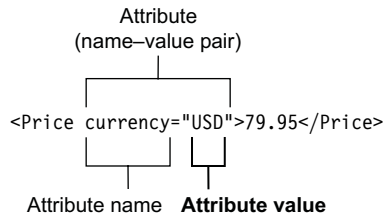**Figure A.19 An example of an attribute name.**

Figure A.20 An example of an attribute value.

Attribute node One of the seven NODE TYPES of the XPATH DATA MODEL that represents an ATTRIBUTE. Note that in XPATH, an ELEMENT NODE is the PARENT of an ATTRIBUTE NODE, but an attribute node is *not* the CHILD of its parent ELEMENT NODE.

Attribute specification A term that is sometimes used to refer to the individual listing for an ATTRIBUTE in an ATTLIST DECLARATION.

Attribute value In XML, a value assigned to an ATTRIBUTE. An attribute value must be enclosed in either double (" ") or single (' ') quotation marks and can contain any CHARACTER DATA, including WHITE SPACE. See Figure A.20. *See also* ATTRIBUTE VALUE; NAME–VALUE PAIR; STRING LITERAL.

attributeGroup element In an XML SCHEMA, a special ELEMENT that is used as an INDICATOR that declares a named group of ATTRIBUTES, as shown in Figure A.21.

attributeGroup indicator *See* INDICATOR.

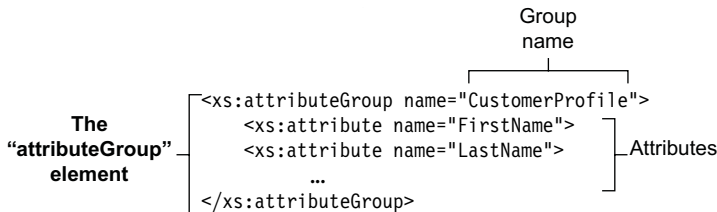**Attribute-list declaration** *See* ATTLIST DECLARATION.

Figure A.21 The "attributeGroup" element.

Authentication In XML messaging, a security term that refers to the possibility to sign a message in a special way, to ensure that the person you are communicating with is indeed that person and that the message has not been change after having been signed. *See also* XML SIGNATURE.

Authentication and Authorization Extensible Markup Language *See* AUTHXML.

AuthXML (Authentication and Authorization Extensible Markup Language) AuthXML is a standard for encoding authentication and authorization information in transport-independent XML. (The "Auth" stands for both authentication and authorization). More details of AuthXML may be obtained from `http://www.rsasecurity.com/`.

Avatar Markup Language *See* AML.

Axes The plural of AXIS.

Axis In XPATH, the first part of a LOCATION STEP that specifies a NODE SET relative to the CONTEXT NODE. It defines the NODE TREE relationship between the context node and one or more NODES that are selected by the location step as candidates for the next context node. There are 13 different axes: (1) SELF AXIS; (2) CHILD AXIS; (3) DESCENDANT AXIS; (4) DESCENDANT-OF-SELF AXIS; (5) PARENT AXIS; (6) ANCESTOR AXIS; (7) ANCESTOR-OF-SELF AXIS; (8) FOLLOWING-SIBLING AXIS; (9) PRECEDING-SIBLING AXIS; (10) FOLLOWING AXIS; (11) PRECEDING AXIS; (12) ATTRIBUTE AXIS; (13) NAMESPACE AXIS.

AXML (Article Extensible Markup Language) An XML-based language for electronic markup of pages intended for hard copy. Details of AXML can be found at `http://xml.gsfc.nasa.gov/article/`.