XML Metadata Interchange (XMI) Appendices

Proposal to the OMG OA&DTF RFP 3:
Stream-based Model Interchange Format (SMIF)

Joint Submission

Cooperative Research Centre for Distributed Systems Technology (DSTC)

International Business Machines Corporation

Oracle Corporation

Platinum Technology, Inc.

Unisys Corporation

Supported by:

Cayenne Software
Genesis Development
Inline Software
Rational Software Corporation
Select Software
Sprint Communications Company
Sybase, Inc.

OMG Document ad/98-07-03
July 6, 1998

Table of Contents

UML 1.1 DTD A

A.1 Introduction

Appendix A contains a DTD generated by hand that represents the UML metamodel. This DTD generally follows the specification of the above section on representing metamodel information. By examining this DTD, you can gain a better understanding of the types of metamodel information that can be represented in an XML DTD, and the information that cannot be specified.

The structure of the DTD closely corresponds to the document “UML Semantics version 1.1, 1 September 1997”. Each XML element corresponding to a class has a comment indicating which pages of that document describe the class. You can verify the accuracy of the DTD against the document by reading the pages of the document in the comments and verifying that the encoding for them is correct.

The DTD is organized according to the packages in the UML metamodel. For example, the Core package is presented first.

A DTD automatically generated from the MOF for UML using the Hierarchical Entity DTD generation rules (Rule Set 3) should closely resemble the example DTD, except that the example DTD uses an additional level of entity definition for elementary items such as attributes.

Considering the issues that arose from representing UML in an XML DTD, aided the development of this specification.

The UML DTD sample can also be used by tools which exchange UML information as a standard for importing and exporting UML metamodels. It can be used for that purpose even if the tools do not directly deal with the MOF.

Note that the UML DTD covers the UML semantics but not the UML notation. Additional work may address the issue of the UML diagrammatic information as an optional level of interchange.

A.2 UML DTD

```
<?xml version="1.0" encoding="utf-8" ?>

<!-- UML Model Document Type Declaration -->
<!-- IBM                                  -->

<!-- Version: 1.23                        -->
<!-- Date: 7/3/98                         -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- The general design of the DTD closely follows the UML description  -->
<!-- in the UML Semantics version 1.1 document dated September 1, 1997. -->

<!-- The first section of this DTD defines XMI transfer text XML         -->
<!-- elements for packaging UML information and transmitting in the XMI -->
<!-- format.  The rest of the DTD defines the UML metamodel.             -->

<!-- It is the intent of this DTD to define a XML ELEMENT for each      -->
<!-- class in the UML metamodel that is not abstract.  The content      -->
<!-- models for each element enforce the UML metamodel containment      -->
<!-- hierarchy.                                                         -->

<!-- Each attribute in a UML metamodel class is represented by an XML   -->
<!-- element.  If the attribute is an enumerated type, the content of   -->
<!-- the XML element is EMPTY, and there is an attribute with the name  -->
<!-- "value" defined for the XML element which holds the literal.  If   -->
<!-- attribute is not an enumerated type, its corresponding XML element -->
<!-- has character data content.                                        -->

<!-- This DTD also defines XML ENTITIES to represent the attributes,    -->
<!-- associations, and compositions for classes in the UML metamodel.   -->
<!-- The attributes in UML metamodel classes are represented by         -->
<!-- entities with a suffix of "Properties".  The associations are      -->
<!-- represented by entities with a suffix of "Associations".           -->
<!-- Containment is represented by entities with a suffix of            -->
<!-- "Compositions".  For all entities comprising the definition of     -->
<!-- UML metamodel classes, the name of the entity is prefixed with the -->
<!-- name of the class from the UML metamodel.  For example, the        -->
<!-- UML metamodel class "ModelElement" is represented by three         -->
<!-- entities:  "ModelElementProperties", "ModelElementAssociations",   -->
<!-- and "ModelElementCompositions".                                    -->

<!-- The associations and attributes of association classes in the UML  -->
<!-- metamodel are reflected in this DTD by moving them to an entity    -->
<!-- corresponding to a UML metamodel abstract class or an element      -->
<!-- corresponding to a UML metamodel concrete class.  Hence, there are -->
<!-- no elements in this DTD corresponding to association classes in    -->
<!-- the UML metamodel.                                                 -->

<!-- Each association role in the UML metamodel is implemented with a   -->
<!-- reference element defined below.  The rationale for                -->
<!-- implementing a reference element is to provide a means for XML     -->
<!-- browsers to enable users to traverse models by following links     -->
```

```
<!-- between related model elements.  The reference element may be       -->
<!-- redefined later to employ more sophisticated XML links than the     -->
<!-- simple internal link implemented here.  Note that it is not         -->
<!-- possible to enforce which kind of elements are referenced by the    -->
<!-- reference element in the DTD itself, so XML documents that are      -->
<!-- valid according to this DTD may be semantically incorrect.          -->

<!-- Since XML does not currently provide a means of inheriting          -->
<!-- attributes and element content between elements, inheritance in     -->
<!-- the UML metamodel is implemented by including the entities which    -->
<!-- define superclasses in the definition of the entities which define -->
<!-- subclasses.                                                           -->

<!-- Where multiple inheritance occurs in the UML metamodel, the         -->
<!-- inherited attributes and content of elements in this DTD are only   -->
<!-- included once in the declaration of the element corresponding to    -->
<!-- a subclass.                                                           -->

<!-- This DTD is organized in sections which correspond to the packages -->
<!-- in the UML metamodel.                                                 -->

<!-- There has been no effort to be concise in defining ELEMENT names.   -->
<!-- Clarity was more of a priority than shortening the length of XML    -->
<!-- documents that are valid according to this DTD.                       -->

<!-- The element declarations are arranged according to the packages in -->
<!-- the UML metamodel.  For each package, the entities which             -->
<!-- correspond to abstract classes in the package are declared.  The    -->
<!-- entities are arranged in alphabetical order by the abstract class   -->
<!-- names.  Then the elements corresponding to concrete UML metamodel   -->
<!-- classes are listed in alphabetical order.  Please note that some    -->
<!-- classes are in multiple packages of the UML metamodel.  The         -->
<!-- complete declaration appears in the first package, and parts of     -->
<!-- the class defined in other packages are included in the              -->
<!-- declarations of those other classes.                                  -->

<!-- Known deviations from UML are noted where they occur, with the      -->
<!-- phrase “UML DEVIATION”                                                -->

<!-- If there are inconsistencies in the UML Semantics document, that    -->
<!-- affect this DTD, they are noted with the phrase “UML                 -->
<!-- INCONSISTENCY”.                                                       -->

<!-- Tags that correspond to UML classes have a name that starts with    -->
<!-- a capital letter.  Tags that correspond to associations in          -->
<!-- the UML metamodel are in lower case letters.                          -->

<!-- This DTD is basically organized according to the chapters in the    -->
<!-- UML Semantics document.  The abstract classes in each chapter are   -->
<!-- listed first, followed by the concrete classes.  The classes are    -->
<!-- in order from superclasses to subclasses, since the definition of   -->
<!-- a subclass requires all of its superclasses to be defined.          -->

<!-- All page and chapter numbers in this documentation refer to the     -->
<!-- document “UML Semantics version 1.1, 1 September 1997”.             -->
```

```
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- This section defines the XMI transfer text XML elements.             -->
<!-- UML DEVIATION:  None of these elements are derived from the UML      -->
<!-- metamodel.  They are defined by the XMI submission in response to    -->
<!-- the SMIF RFP from OMG.                                               -->
<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->


<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- XMI is the top-level XML element for XMI transfer text.      -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI (XMI.header, XMI.content, XMI.extensions*) >
<!ATTLIST XMI
              xmi-version CDATA #FIXED “1.0”
              timestamp   CDATA #IMPLIED
              verified (true | false) #IMPLIED
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- XMI.header contains metadata about the transfer and data about       -->
<!-- the metamodels which define the content of the transfer.             -->
<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.header (XMI.documentation?, XMI.metamodel+) >

<!-- ____________________________________________________________________ -->
<!--                                                                      -
->
<!-- documentation contains data about the information being transferred.  -->
<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.documentation (#PCDATA |
                             XMI.owner | XMI.contact |
                             XMI.longDescription |
                             XMI.shortDescription | XMI.exporter |
                             XMI.exporterVersion | XMI.notice)* >

<!ELEMENT XMI.owner ANY >
<!ELEMENT XMI.contact ANY >
<!ELEMENT XMI.longDescription ANY >
<!ELEMENT XMI.shortDescription ANY >
<!ELEMENT XMI.exporter ANY >
<!ELEMENT XMI.exporterVersion ANY >
<!ELEMENT XMI.exporterID ANY >
<!ELEMENT XMI.notice ANY >

<!-- ____________________________________________________________________ -->
```

```
<!--                                                                    -
->
<!-- metamodel contains data about the metamodel(s) to which the content   -->
<!-- conforms.                                                             -
->
<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.metamodel ANY>
<!ATTLIST XMI.metamodel
                name     CDATA #REQUIRED
                version CDATA #REQUIRED
                href     CDATA #IMPLIED
>

<!-- _____________________________________________ -->
<!--                                                 -->
<!-- content is the actual data being transferred.   -->
<!-- _____________________________________________ -->

<!ELEMENT XMI.content ANY >

<!-- ____________________________________________________________ -->
<!--                                                               -->
<!-- extensions contains information related to the content that is not  -->
<!-- defined in the metamodel(s) in the header.                    -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.extensions ANY >

<!-- ____________________________________________________________ -->
<!--                                                               -->
<!-- This section contains the XMI.reference element declaration.  It   -->
<!-- can be either an internal reference, in which case the target       -->
<!-- attribute contains the ID of the XML element being referred to,     -->
<!-- or it can be a reference to an XML element in another document,     -->
<!-- in which case the href attribute holds the URI of the document      -->
<!-- with an XPointer at the end.  When the XLink working draft becomes -->
<!-- a W3C recommendation, this element will be redefined to be a        -->
<!-- simple XLink.                                                       -->

<!-- The expectedType attribute may contain the expected type of the     -->
<!-- XML element to be referred to.  Although XML processors will not    -->
<!-- enforce that the XML element referred to is the expected type,      -->
<!-- tools may report a warning if the expectedType does not match       -->
<!-- what is being referred to.                                          -->

<!-- UML DEVIATION:  The XMI.reference element is not defined in UML.It -->
<!-- is used to implement associations between UML metamodel classes.    -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI.reference ANY >
<!ATTLIST XMI.reference
               target IDREF #IMPLIED
               href    CDATA #IMPLIED
               expectedType CDATA #IMPLIED
```

```
                content-title CDATA #IMPLIED
>

<!-- _____________________________________________________________________ -->
<!--                                                                       -->
<!-- This section contains the data types which are defined in the MOF. -->
<!-- _____________________________________________________________________ -->

<!ELEMENT XMI.field ANY >
<!ELEMENT XMI.struct (field)+ >

<!ELEMENT XMI.seqItem ANY >
<!ELEMENT XMI.sequence (seqItem)* >

<!ELEMENT XMI.arrayLen ANY >
<!ELEMENT XMI.arrayItem ANY >
<!ELEMENT XMI.array (XMI.arrayLen, XMI.arrayItem*) >

<!ELEMENT XMI.enum (#PCDATA) >

<!ELEMENT XMI.discrim ANY >
<!ELEMENT XMI.union (XMI.discrim, XMI.field*) >

<!ELEMENT XMI.any ANY >
<!ATTLIST XMI.any
              type CDATA #IMPLIED
>

<!-- _____________________________________________________________________ -->
<!-- _____________________________________________________________________ -->
<!--                                                                       -->
<!--                           CHAPTER 7                                   -->
<!-- This section contains the declarations of the entities and          -->
<!-- elements corresponding to the Data Types package, one of the        -->
<!-- foundation packages in the UML metamodel.  The UML standard         -->
<!-- enumerations are defined here, because they are used to define      -->
<!-- the rest of UML, and XML entities must be declared before being     -->
<!-- used in a DTD.                                                      -->
<!-- _____________________________________________________________________ -->
<!-- _____________________________________________________________________ -->

<!ENTITY % AggregationKind
           ‘ XMI.value (none | shared | composite) #REQUIRED ‘ >

<!ENTITY % Boolean
           ‘ XMI.value (true|false) #REQUIRED ‘ >

<!ENTITY % CallConcurrencyKind
           ‘ XMI.value (sequential | guarded | concurrent) #REQUIRED ‘ >

<!ENTITY % ChangeableKind
           ‘ XMI.value (none | frozen | addOnly) #REQUIRED ‘ >

<!ENTITY % ParameterDirectionKind
           ‘ XMI.value (in | out | inout | return) #REQUIRED ‘ >
```

```
<!ENTITY % PseudoStateKind
          ‘ XMI.value (initial | deepHistory | shallowHistory |
                    join | fork | branch | final) #REQUIRED ‘ >

<!ENTITY % ScopeKind
          ‘ XMI.value (classifier | instance) #REQUIRED ‘ >

<!ENTITY % SynchronousKind
          ‘ XMI.value (synchronous | asynchronous) #REQUIRED ‘ >

<!ENTITY % VisibilityKind
          ‘ XMI.value (public | protected |private) #REQUIRED ‘ >

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                           CHAPTER 4                                  -->
<!-- This section contains the declarations of the entities and           -->
<!-- elements corresponding to the core package, one of the foundation    -->
<!-- packages in the UML metamodel.                                       -->
<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->


<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- The declaration of the entities in this section correspond to the    -->
<!-- abstract classes in the core package.                                -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASSES:  ModelElement and ElementOwnership                      -->

<!-- The definition of the ModelElement class is on pages 16, 17, 25,    -->
<!-- and 32.  Other aspects of the ModelElement class are documented in   -->
<!-- later sections of this DTD and the semantics document.               -->

<!-- The definition of the ElementOwnership class is on pages 16, 22,    -->
<!-- and 31.                                                              -->

<!-- These are properties common to all elements.  Note that the name    -->
<!-- attribute is from the ModelElement class, the visibility attribute   -->
<!-- is from the ElementOwnership class, and the id attribute is not      -->
<!-- defined by UML, but is required to implement UML associations        -->
<!-- using the XMI.reference XML ELEMENT defined above.                   -->

<!-- The content common to every class that inherits from the             -->
<!-- ModelElement class is included here.  Some of the content is in      -->
<!-- other packages which will be defined later.                          -->

<!-- UML DEVIATION:  The visibility attribute is implemented in this      -->
<!-- DTD as an attribute in the ModelElement class; in UML, it is an      -->
<!-- attribute of the ElementOwnership class.                             -->
```

```
<!-- UML DEVIATION:  The visibility attribute is from the                -->
<!-- ElementOwnership class, an association class in the UML metamodel. -->

<!-- UML INCONSISTENCY: The multiplicity of the stereotype association  -->
<!-- end is * in the diagram on p. 53 but is documented as 0 or 1 on    -->
<!-- page 54.                                                           -->
<!-- ___________________________________________________________________ -->


<!ENTITY % modelElementName 'name' >
<!ELEMENT name (#PCDATA) >

<!ENTITY % modelElementVisibility 'visibility' >
<!ELEMENT visibility EMPTY >
<!ATTLIST visibility %VisibilityKind; >

<!ENTITY % ModelElementProperties '%modelElementName;,
                                   %modelElementVisibility;' >

<!-- ___________________________________________________________________ -->
<!--                                                                     -->
<!-- namespace should contain a reference to a model element that        -->
<!-- corresponds to an instance of the Namespace UML class or one of     -->
<!-- its many subclasses.                                                -->
<!-- ___________________________________________________________________ -->

<!ENTITY % modelElementNamespace 'namespace' >
<!ELEMENT namespace (XMI.reference) >

<!ENTITY % modelElementStereotype 'stereotype' >
<!ELEMENT stereotype (XMI.reference) >

<!ENTITY % modelElementConstraint 'constraint' >
<!ELEMENT constraint (XMI.reference) >

<!-- ___________________________________________________ -->
<!--                                                     -->
<!-- view should contain a reference to a ViewElement    -->
<!-- ___________________________________________________ -->

<!ENTITY % modelElementView 'view' >
<!ELEMENT view (XMI.reference) >

<!-- ___________________________________________________________________ -->
<!--                                                                     -->
<!-- provision can contain a reference to any ModelElement.              -->
<!-- ___________________________________________________________________ -->

<!ENTITY % modelElementProvision 'provision' >
<!ELEMENT provision (XMI.reference) >

<!-- ___________________________________________________________________ -->
<!--                                                                     -->
<!-- requirement can contain a reference to any ModelElement.            -->
```

```
<!-- ____________________________________________________________________ -->

<!ENTITY % modelElementRequirement 'requirement' >
<!ELEMENT requirement (XMI.reference) >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- templateParameter should contain a reference to a Parameter.  -->
<!-- ____________________________________________________________________ -->

<!ENTITY % modelElementTemplateParameter 'templateParameter' >
<!ELEMENT templateParameter (XMI.reference) >

<!-- _____________________________________________ -->
<!--                                                -->
<!-- template should refer to a model element  -->
<!-- _____________________________________________ -->

<!ENTITY % modelElementTemplate 'template' >
<!ELEMENT template (reference) >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- extensions may refer to an XML element which contains          -->
<!-- information that is not defined in the metamodel.              -->
<!-- UML DEVIATION:  This is not defined in UML; it is one XMI      -->
<!-- technique for representing model information that extends a   -->
<!-- metamodel.                                                           -->
<!-- ____________________________________________________________________ -->

<!ENTITY % modelElementExtension 'XMI.extension' >
<!ELEMENT XMI.extension ANY >

<!ENTITY % ModelElementAssociations '%modelElementNamespace;?,
                                     %modelElementTemplate;?,
                                     (%modelElementStereotype; |
                                      %modelElementConstraint; |
                                      %modelElementView; |
                                      %modelElementProvision; |
                                      %modelElementRequirement; |
                                      %modelElementTemplateParameter; |
                                      %modelElementExtension;)*' >

<!ENTITY % taggedValue 'taggedValue' >
<!ELEMENT taggedValue (TaggedValue) >

<!ENTITY % modelElementBehavior 'behavior' >
<!ELEMENT behavior (StateMachine | ActivityModel) >

<!ENTITY % remoteContent 'XMI.remoteContent' >
<!ELEMENT XMI.remoteContent (XMI.reference) >

<!ENTITY % ModelElementCompositions '(%taggedValue; |
                                      %modelElementBehavior;)*' >
```

```
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Feature                                                   -->

<!-- The definition of the Feature class is on pages 16, 23, and 31.     -->

<!-- UML DEVIATION:  The visibility attribute in the Feature class is    -->
<!-- not included here, it comes from the ModelElementProperties entity -->
<!-- defined below. The name attribute is also included in the           -->
<!-- ModelElementProperties entity defined below.                        -->
<!-- ____________________________________________________________________ -->


<!ENTITY % ownerScope 'ownerScope' >
<!ELEMENT ownerScope EMPTY >
<!ATTLIST ownerScope %ScopeKind; >

<!ENTITY % FeatureProperties '%ModelElementProperties;,
                                  %ownerScope;' >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- owner refers to the Classifier which the feature belongs to          -->
<!-- ____________________________________________________________________ -->

<!ENTITY % featureOwner 'owner' >
<!ELEMENT owner (XMI.reference) >

<!ENTITY % FeatureAssociations '%ModelElementAssociations;,
                                    %featureOwner;' >

<!ENTITY % FeatureCompositions '%ModelElementCompositions;' >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: BehavioralFeature                                         -->

<!-- The definition of the BehavioralFeature class is on pages 16, 20,   -->
<!-- and 28, 66.                                                          -->

<!-- UML DEVIATION:  The attribute name from the UML class               -->
<!-- BehavioralFeature is not included here because there is a name      -->
<!-- attribute for every model element defined in the elementProperties -->
<!-- entity declared below.                                               -->
<!-- ____________________________________________________________________ -->

<!ENTITY % isQuery 'isQuery' >
<!ELEMENT isQuery EMPTY >
<!ATTLIST isQuery %Boolean; >

<!ENTITY % BehavioralFeatureProperties '%FeatureProperties;,
                                            %isQuery;' >

<!ENTITY % raisedException 'raisedException' >
<!ELEMENT raisedException (XMI.reference) >
```

```
<!ENTITY % BehavioralFeatureAssociations '%FeatureAssociations;,
                                          %raisedException;*' >

<!ENTITY % parameter 'parameter' >
<!ELEMENT parameter (Parameter) >

<!ENTITY % BehavioralFeatureCompositions '%FeatureCompositions;,
                                         (parameter | StateMachine |
                                          ActivityModel)*' >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Namespace                                                 -->

<!-- The definition of the Namespace class is on pages 16, 17, 25, 26,   -->
<!-- and 33.                                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % NamespaceProperties '%ModelElementProperties;' >

<!ENTITY % NamespaceAssociations '%ModelElementAssociations;' >

<!ENTITY % modelElements '(Model |
                           Subsystem |
                           Package |
                           Class |
                           Generalization |
                           Association |
                           Interface |
                           ViewElement |
                           Usecase |
                           AssociationClass |
                           Node |
                           Component |
                           Comment |
                           Signal |
                           Exception |
                           Object |
                           Link |
                           LinkObject |
                           UsecaseInstance |
                           ElementReference |
                           Actor |
                           Dependency |
                           Refinement |
                           Usage |
                           Trace |
                           Binding |
                           Constraint |
                           Stereotype |
                           StateMachine |
                           ActivityModel |
                           Collaboration |
                           Request |
```

```
                                ClassifierRole |
                                AssociationRole)' >

<!ENTITY % ownedElement 'ownedElement' >
<!ELEMENT ownedElement %modelElements; >

<!ENTITY % NamespaceCompositions '(%ModelElementCompositions;,
                                        %ownedElement;*) ' >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: GeneralizableElement                                      -->

<!-- The definition of the GeneralizableElement class is on pages 16,    -->
<!-- 17, 23, 24, 31, and 32.                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % isAbstract 'isAbstract' >
<!ELEMENT isAbstract EMPTY >
<!ATTLIST isAbstract %Boolean; >

<!ENTITY % isLeaf 'isLeaf' >
<!ELEMENT isLeaf EMPTY >
<!ATTLIST isLeaf %Boolean; >

<!ENTITY % isRoot 'isRoot' >
<!ELEMENT isRoot EMPTY >
<!ATTLIST isRoot %Boolean; >

<!ENTITY % GeneralizableElementProperties '%NamespaceProperties;,
                                           %isAbstract;,
                                           %isLeaf;,
                                           %isRoot;' >

<!-- _____________________________________________________________ -->
<!--                                                                -->
<!-- Both generalization and specialization should refer to     -->
<!-- GeneralizableElements.                                         -->
<!-- _____________________________________________________________ -->

<!ENTITY % generalization 'generalization' >
<!ELEMENT generalization (XMI.reference) >

<!ENTITY % specialization 'specialization' >
<!ELEMENT specialization (XMI.reference) >

<!ENTITY % GeneralizableElementAssociations '%NamespaceAssociations;,
                                             (%generalization; |
                                              %specialization;)*' >

<!ENTITY % GeneralizableElementCompositions '%NamespaceCompositions;' >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Classifier                                                -->
```

```
<!-- The definition of the Classifier class is on pages 16, 17, 21, 29  -->
<!-- and 30.                                                             -->

<!-- NOTE:  Classifiers can contain state machines according to          -->
<!-- page 102.                                                           -->

<!-- ____________________________________________________________________ -->

<!ENTITY % ClassifierProperties '%GeneralizableElementProperties;' >

<!-- ____________________________________________________________________ -->
<!--                                                                     -->
<!-- participant should contain a reference to an AssociationEnd         -->
<!-- ____________________________________________________________________ -->

<!ENTITY % participant 'participant' >
<!ELEMENT participant (XMI.reference) >

<!-- ____________________________________________________________________ -->
<!--                                                                     -->
<!-- realization should contain a reference to a Classifier, except for -->
<!-- Interfaces.                                                         -->
<!-- ____________________________________________________________________ -->

<!ENTITY % realization 'realization' >
<!ELEMENT realization (XMI.reference) >

<!-- ____________________________________________________________________ -->
<!--                                                                     -->
<!-- specification should contain a reference to a Classifier.           -->
<!-- Methods also have a specification reference, which should contain   -->
<!-- a reference to the Operation the method implements.                 -->
<!-- ____________________________________________________________________ -->

<!ENTITY % specification 'specification' >
<!ELEMENT specification (XMI.reference) >

<!ENTITY % ClassifierAssociations '%GeneralizableElementAssociations;,
                                   (%participant; |
                                    %realization; |
                                    %specification;)*' >

<!ENTITY % ClassifierCompositions '%GeneralizableElementCompositions;,
                                   feature*' >

<!ELEMENT feature (Attribute |
                   Operation |
                   Method |
                   Reception) >

<!-- ____________________________________________________________________ -->
<!--                                                                     -->
<!-- UML CLASS: Element                                                  -->
```

```
<!-- The definition of the Element class is on pages 16, 22, and 31.     -->

<!-- UML DEVIATION:  The id attribute is not defined in UML; it is used -->
<!-- to implement associations in the UML metamodel in XML.               -->
<!-- ____________________________________________________________________ -->

<!ENTITY % XMI.ElementAttributes 'XMI.id ID #REQUIRED
                                        XMI.remote (true | false) "false" ' >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ElementOwnership                                          -->

<!-- The definition of the ElementOwnership class is on pages 16, 22,    -->
<!-- and 31.                                                              -->

<!-- UML DEVIATION:  The ElementOwnership attribute visibility is        -->
<!-- in this DTD in the ModelElementProperties entity. This DTD treats   -->
<!-- the visibility attribute as an attribute of every ModelElement.     -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________  -->
<!--                                                                       -->
<!-- UML CLASS: StructuralFeature                                          -->

<!-- The definition of the StructuralFeature class is on pages 16, 27,    -->
<!-- and 34.                                                              -->

<!-- The declaration of the type element is with the declaration of the   -->
<!-- Attribute element.                                                   -->
<!-- ____________________________________________________________________  -->

<!ENTITY % changeable 'changeable' >
<!ELEMENT changeable EMPTY >
<!ATTLIST changeable %ChangeableKind; >

<!ENTITY % multiplicity 'multiplicity' >
<!ELEMENT multiplicity (#PCDATA) >

<!ENTITY % StructuralFeatureProperties '%FeatureProperties;,
                                        %changeable;,
                                        %multiplicity;' >

<!ENTITY % structuralFeatureType 'type' >
<!ELEMENT type (#PCDATA |
                  XMI.reference |
                  XMI.struct |
                  XMI.sequence |
                  XMI.enum |
                  XMI.array |
                  XMI.union |
                  XMI.any)*
>

<!ENTITY % StructuralFeatureAssociations '%FeatureAssociations;,
```

```
                                                    %structuralFeatureType;' >

<!ENTITY % StructuralFeatureCompositions '%FeatureCompositions;' >

<!-- ____________________________________________________________________  -->
<!--                                                                        -->
<!-- The element declarations in this section correspond to the concrete -->
<!-- classes in the core package.                                           -->
<!-- ____________________________________________________________________  -->

<!-- ____________________________________________________________________  -->
<!--                                                                        -->
<!-- UML CLASS: Association                                                 -->

<!-- The definition of the Association class is on pages 17, 27, and 28. -->

<!-- UML DEVIATION:  Since two or more association ends are contained    -->
<!-- within each association, the two required ends are represented as   -->
<!-- source and target elements in this DTD. They are not defined in     -->
<!-- UML.                                                                  -->
<!-- ____________________________________________________________________  -->

<!ENTITY % AssociationProperties '%GeneralizableElementProperties;' >

<!ENTITY % AssociationAssociations '%GeneralizableElementAssociations;' >

<!ENTITY % source 'source' >
<!ELEMENT source (AssociationEnd) >

<!ENTITY % target 'target' >
<!ELEMENT target (AssociationEnd) >

<!ENTITY % connection 'connection' >
<!ELEMENT connection (AssociationEnd) >

<!ENTITY % associationEnds "%source;, %target;, %connection;*" >

<!ENTITY % AssociationCompositions '%GeneralizableElementCompositions;,
                                        %associationEnds;' >

<!ELEMENT Association (%remoteContent; |
                           (%AssociationProperties;,
                            %AssociationAssociations;,
                            %AssociationCompositions;)) >
<!ATTLIST Association
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________  -->
<!--                                                                        -->
<!-- UML CLASS: Class                                                       -->

<!-- The definition of the Class class is on pages 16, 20, 21, and 29.    -->

<!-- NOTE:  The content of a class is not completely defined in the core -->
```

```
<!-- package.  Some of it is defined in other UML metamodel packages.     -->
<!-- ____________________________________________________________________  -->

<!ENTITY % isActive 'isActive' >
<!ELEMENT isActive EMPTY >
<!ATTLIST isActive %Boolean; >

<!ENTITY % ClassProperties '%ClassifierProperties;,
                                %isActive;' >

<!ENTITY % ClassAssociations '%ClassifierAssociations;' >

<!ENTITY % ClassCompositions '%ClassifierCompositions;' >

<!ELEMENT Class (%remoteContent; |
                     (%ClassProperties;,
                      %ClassAssociations;,
                      %ClassCompositions;)) >
<!ATTLIST Class
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________  -->
<!--                                                                       -->
<!-- UML CLASS: AssociationClass                                           -->

<!-- The definition of the Association class is on pages 17, 18, and 28. -->

<!-- UML DEVIATION:  Since two or more association ends are contained     -->
<!-- within each association, the two required ends are represented as    -->
<!-- source and target elements in this DTD. They are not defined in      -->
<!-- UML.                                                                  -->
<!-- ____________________________________________________________________  -->

<!ENTITY % AssociationClassProperties '%ClassProperties;' >

<!ENTITY % AssociationClassAssociations '%ClassAssociations;' >

<!ENTITY % AssociationClassCompositions '%ClassCompositions;,
                                             %associationEnds;' >

<!ELEMENT AssociationClass (%remoteContent; |
                                    (%AssociationClassProperties;,
                                     %AssociationClassAssociations;,
                                     %AssociationClassCompositions;)) >
<!ATTLIST AssociationClass
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________  -->
<!--                                                                       -->
<!-- UML CLASS: AssociationEnd                                             -->

<!-- The definition of the Association class is on pages 17, 18, 19, and -->
<!-- 28.                                                                   -->
```

```
<!-- NOTE: Element specification is declared in the declaration of        -->
<!-- abstract class Classifier. Element type is declared with element    -->
<!-- Attribute, element isChangeable is declared with StructuralFeature, -->
<!-- element multiplicity is declared with StructuralFeature, and        -->
<!-- element targetScope is declared with StructuralFeature.             -->

<!-- ____________________________________________________________________  -->


<!ENTITY % isNavigable 'isNavigable' >
<!ELEMENT isNavigable EMPTY >
<!ATTLIST isNavigable %Boolean; >

<!ENTITY % isOrdered 'isOrdered' >
<!ELEMENT isOrdered EMPTY >
<!ATTLIST isOrdered %Boolean; >

<!ENTITY % aggregation 'aggregation' >
<!ELEMENT aggregation EMPTY >
<!ATTLIST aggregation %AggregationKind; >

<!ENTITY % targetScope 'targetScope' >
<!ELEMENT targetScope EMPTY >
<!ATTLIST targetScope %ScopeKind; >

<!ENTITY % AssociationEndProperties '%ModelElementProperties;,
                                     %isNavigable;,
                                     %isOrdered;,
                                     %aggregation;,
                                     %multiplicity;,
                                     %changeable;,
                                     %targetScope;' >


<!-- _____________________________________________________________  -->
<!--                                                                 -->
<!-- qualifier should contain an Attribute or a reference to    -->
<!-- an attribute.                                                  -->
<!-- _____________________________________________________________  -->

<!ENTITY % qualifier 'qualifier' >
<!ELEMENT qualifier (Attribute) >

<!ENTITY % AssociationEndAssociations '%ModelElementAssociations;,
                                       %structuralFeatureType;,
                                       %specification;* ' >

<!ENTITY % AssociationEndCompositions '%ModelElementCompositions;,
                                       %qualifier;* ' >

<!ELEMENT AssociationEnd (%remoteContent; |
                          (%AssociationEndProperties;,
                           %AssociationEndAssociations;,
                           %AssociationEndCompositions;)) >
```

```
<!ATTLIST AssociationEnd
               %XMI.ElementAttributes;
>
<!-- ____________________________________________________________________  -->
<!--                                                                        -->
<!-- UML CLASS: Attribute                                                   -->

<!-- The definition of the Attribute class is on pages 16, 19, 20, and    -->
<!-- 28.                                                                    -->
<!-- ____________________________________________________________________  -->

<!ENTITY % initialValue 'initialValue' >
<!ELEMENT initialValue (#PCDATA) >

<!ENTITY % AttributeProperties '%StructuralFeatureProperties;,
                                    %initialValue;' >

<!ENTITY % associationEnd 'associationEnd' >
<!ELEMENT associationEnd (XMI.reference) >

<!ENTITY % AttributeAssociations '%StructuralFeatureAssociations;,
                                      %associationEnd;?' >

<!ENTITY % AttributeCompositions '%StructuralFeatureCompositions;' >

<!ELEMENT Attribute (%remoteContent; |
                           (%AttributeProperties;,
                            %AttributeAssociations;,
                            %AttributeCompositions;)) >
<!ATTLIST Attribute
              %XMI.ElementAttributes;
>


<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Constraint                                                -->

<!-- The definition of the Constraint class is on pages 16, 21, 22,   -->
<!-- and 31, 53.                                                          -->
<!-- ____________________________________________________________________ -->

<!ENTITY % body 'body' >
<!ELEMENT body (#PCDATA) >
<!ATTLIST body
              xml-space (default | preserve) "preserve" >

<!ENTITY % ConstraintProperties '%ModelElementProperties;,
                                    %body;' >

<!-- ____________________________________________________________________  -->
<!--                                                                       -->
<!-- constrainedElement may contain a reference to any ModelElement. -->
<!-- ____________________________________________________________________  -->
```

```
<!ENTITY % constrainedElement 'constrainedElement' >
<!ELEMENT constrainedElement (XMI.reference) >

<!ENTITY % constrainedStereotype 'constrainedStereotype' >
<!ELEMENT constrainedStereotype (XMI.reference) >

<!ENTITY % ConstraintAssociations '%ModelElementAssociations;,
                                   (%constrainedElement;+ |
                                    %constrainedStereotype;?)' >

<!ENTITY % ConstraintCompositions '%ModelElementCompositions;' >

<!ELEMENT Constraint (%remoteContent; |
                      (%ConstraintProperties;,
                       %ConstraintAssociations;,
                       %ConstraintCompositions;)) >
<!ATTLIST Constraint
          %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: DataType                                                  -->

<!-- ____________________________________________________________________ -->

<!ENTITY % DataTypeProperties '%ClassifierProperties;' >

<!ENTITY % DataTypeAssociations '%ClassifierAssociations;' >

<!ENTITY % DataTypeCompositions '%ClassifierCompositions;' >

<!ELEMENT DataType (%remoteContent; |
                    (%DataTypeProperties;,
                     %DataTypeAssociations;,
                     %DataTypeCompositions;)) >
<!ATTLIST DataType
          %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Dependency                                                -->

<!-- The definition of the Dependency class is on pages 16, 22, 31, and -->
<!-- 45.                                                                  -->

<!-- NOTE: Refinement, usage, trace, binding and owningDependency are   -->
<!-- declared later in the auxiliary elements package.                    -->
<!-- ____________________________________________________________________ -->

<!ENTITY % dependencyDescription 'description' >
<!ELEMENT description (#PCDATA) >

<!ENTITY % DependencyProperties '%ModelElementProperties;,
```

```
                                        %dependencyDescription;' >

<!ENTITY % owningDependency 'owningDependency' >
<!ELEMENT owningDependency (XMI.reference) >

<!-- _____________________________________________________  -->
<!--                                                        -->
<!-- client may contain a reference to any ModelElement. -->
<!-- _____________________________________________________  -->

<!ENTITY % client 'client' >
<!ELEMENT client (XMI.reference) >

<!-- _____________________________________________________  -->
<!--                                                        -->
<!-- supplier may contain a reference to any ModelElement. -->
<!-- _____________________________________________________  -->

<!ENTITY % supplier 'supplier' >
<!ELEMENT supplier (XMI.reference) >

<!ENTITY % DependencyAssociations '%ModelElementAssociations;,
                                        %owningDependency;?,
                                        (%client; |
                                         %supplier;)*' >

<!ENTITY % subDependencies 'subDependencies' >
<!ELEMENT subDependencies     (Dependency |
                               Refinement |
                               Usage |
                               Trace |
                               Binding)* >

<!ENTITY % DependencyCompositions '%ModelElementCompositions;,
                                        %subDependencies;?' >

<!ELEMENT Dependency (%remoteContent; |
                            (%DependencyProperties;,
                             %DependencyAssociations;,
                             %DependencyCompositions;)) >
<!ATTLIST Dependency
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- UML CLASS: Generalization                                    -->

<!-- The definition of the Generalization class is on pages 17, 24,     -->
<!-- and 32.                                                            -->
<!-- ____________________________________________________________ -->

<!ENTITY % discriminator 'discriminator' >
<!ELEMENT discriminator (#PCDATA) >
```

```
<!ENTITY % GeneralizationProperties '%ModelElementProperties;,
                                    %discriminator;' >

<!-- ____________________________________________________________  -->
<!--                                                               -->
<!-- subType may contain a reference to any GeneralizableElement. -->
<!-- ____________________________________________________________  -->

<!ENTITY % subtype 'subtype' >
<!ELEMENT subtype (XMI.reference) >

<!-- ____________________________________________________________  -->
<!--                                                               -->
<!-- supertype may contain a reference to any GeneralizableElement. -->
<!-- ____________________________________________________________  -->

<!ENTITY % supertype 'supertype' >
<!ELEMENT supertype (XMI.reference) >

<!ENTITY % GeneralizationAssociations '%ModelElementAssociations;,
                                      %subtype;,
                                      %supertype;' >

<!ENTITY % GeneralizationCompositions '%ModelElementCompositions;' >

<!ELEMENT Generalization (%remoteContent; |
                          (%GeneralizationProperties;,
                           %GeneralizationAssociations;,
                           %GeneralizationCompositions;)) >
<!ATTLIST Generalization
          %XMI.ElementAttributes;
>
<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- UML CLASS: Interface                                         -->

<!-- The definition of the Interface class is on pages 16, 24, 25,  -->
<!-- and 32.                                                        -->
<!-- ____________________________________________________________ -->

<!ENTITY % InterfaceProperties '%ClassifierProperties;' >

<!ENTITY % InterfaceAssociations '%ClassifierAssociations;' >

<!ENTITY % InterfaceCompositions '%ClassifierCompositions;' >

<!ELEMENT Interface (%remoteContent; |
                     (%InterfaceProperties;,
                      %InterfaceAssociations;,
                      %InterfaceCompositions;)) >
<!ATTLIST Interface
          %XMI.ElementAttributes;
>

<!-- ____________________________________________________________ -->
```

```
<!--                                                                    -->
<!-- UML CLASS: Method                                                  -->

<!-- The definition of the Method class is on pages 16, 25, and 32.     -->

<!-- NOTE:  The declaration of XML ELEMENTS specification,              -->
<!-- StateMachine and ActivityModel are in other sections of this DTD. -->
<!-- ____________________________________________________________________ -->

<!ENTITY % MethodProperties '%BehavioralFeatureProperties;,
                                %body;' >

<!ENTITY % MethodAssociations '%BehavioralFeatureAssociations;,
                                  %specification;' >

<!ENTITY % MethodCompositions '%BehavioralFeatureCompositions;' >

<!ELEMENT Method (%remoteContent; |
                        (%MethodProperties;,
                         %MethodAssociations;,
                         %MethodCompositions;)) >
<!ATTLIST Method
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                    -->
<!-- UML CLASS: Operation                                               -->

<!-- The definition of the Operation class is on pages 16, 26, and 34. -->

<!-- NOTE:  The declaration of XML ELEMENTS realization,                -->
<!-- stateMachine and activityModel are in other sections of this DTD. -->

<!-- UML DEVIATION:  This DTD uses element "realization" to reference  -->
<!-- the methods that implement this operation.  The association end   -->
<!-- in the UML metamodel does not have a name.                        -->
<!-- ____________________________________________________________________ -->

<!ENTITY % concurrency 'concurrency' >
<!ELEMENT concurrency EMPTY >
<!ATTLIST concurrency %CallConcurrencyKind; >

<!ENTITY % isPolymorphic 'isPolymorphic' >
<!ELEMENT isPolymorphic EMPTY >
<!ATTLIST isPolymorphic %Boolean; >

<!ENTITY % operationSpecification 'operationSpecification' >
<!ELEMENT operationSpecification (#PCDATA) >

<!ENTITY % OperationProperties '%BehavioralFeatureProperties;,
                                    %concurrency;,
                                    %isPolymorphic;,
                                    %operationSpecification;' >
```

```
<!-- _____________________________________________________ -->
<!--                                                       -->
<!-- occurrence should contain a reference to a CallEvent  -->
<!-- _____________________________________________________ -->

<!ENTITY % occurrence 'occurrence' >
<!ELEMENT occurrence (XMI.reference) >

<!ENTITY % OperationAssociations '%BehavioralFeatureAssociations;,
                                    (%realization; |
                                     %occurrence;)*' >

<!ENTITY % OperationCompositions '%BehavioralFeatureCompositions;' >

<!ELEMENT Operation (%remoteContent; |
                           (%OperationProperties;,
                            %OperationAssociations;,
                            %OperationCompositions;)) >
<!ATTLIST Operation
             %XMI.ElementAttributes;
>

<!-- _____________________________________________________________ -->
<!--                                                               -->
<!-- UML CLASS: Parameter                                          -->

<!-- The definition of the Parameter class is on pages 16, 26, 27,  -->
<!-- and 34.                                                       -->

<!-- NOTE:  The declaration of XML ELEMENT type is declared with    -->
<!-- XML ELEMENT attribute.                                        -->

<!-- UML DEVIATION:  The name attribute is declared in              -->
<!-- ModelElementStructure, not with the Parameter class.          -->
<!-- _____________________________________________________________ -->

<!ENTITY % parameterDefaultValue 'defaultValue' >
<!ELEMENT defaultValue (#PCDATA) >

<!ENTITY % parameterKind 'kind' >
<!ELEMENT kind EMPTY >
<!ATTLIST kind %ParameterDirectionKind; >

<!ENTITY % ParameterProperties  '%ModelElementProperties;,
                                    %parameterDefaultValue;,
                                    %parameterKind;' >

<!ENTITY % ParameterAssociations '%ModelElementAssociations;,
                                     %structuralFeatureType;' >

<!ENTITY % ParameterCompositions '%ModelElementCompositions;' >

<!ELEMENT Parameter (%remoteContent; |
                           (%ParameterProperties;,
                            %ParameterAssociations;,
```

```
                                %ParameterCompositions;)) >
<!ATTLIST Parameter
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                          CHAPTER 5                                   -->
<!-- This section contains the declarations of the elements              -->
<!-- corresponding to the auxiliary elements subpackage of the           -->
<!-- foundation package of the UML metamodel.                            -->
<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->


<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- There are no abstract classes in this package.  The concrete       -->
<!-- classes are handled in this section.                                -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Binding                                                   -->

<!-- The definition of the Binding class is on pages 43 and 44.         -->

<!-- UML DEVIATION:  The argument association is implemented as a       -->
<!-- reference instead of containment.                                   -->
<!-- ____________________________________________________________________ -->

<!ENTITY % BindingProperties '%DependencyProperties;' >

<!-- ___________________________________________ -->
<!--                                             -->
<!-- argument may contain any ModelElement   -->
<!-- ___________________________________________ -->

<!ENTITY % argument 'argument' >
<!ELEMENT argument (%modelElements;) >

<!ENTITY % BindingAssociations '%DependencyAssociations;' >

<!ENTITY % BindingCompositions '%DependencyCompositions;,
                                      argument+' >

<!ELEMENT Binding (%remoteContent; |
                          (%BindingProperties;,
                           %BindingAssociations;,
                           %BindingCompositions;)) >
<!ATTLIST Binding
              %XMI.ElementAttributes;
>
<!-- ____________________________________________________________________ -->
```

```
<!--                                                                    -->
<!-- UML CLASS: Comment                                                 -->

<!-- The definition of the Comment class is on pages 44 and 45.         -->

<!-- UML DEVIATION:  The semantics document is ambiguous about whether -->
<!-- comment inherits from ModelElement or ViewElement.  This DTD       -->
<!-- defines it as a ModelElement.                                      -->
<!-- ____________________________________________________________________ -->

<!ENTITY % text 'text' >
<!ELEMENT text (#PCDATA) >
<!ATTLIST text
       xml-space (default | preserve) "preserve"
>

<!ENTITY % CommentProperties '%ModelElementProperties;' >


<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- element may contain a reference to any ModelElement.      -->
<!-- ____________________________________________________________ -->

<!ENTITY % element 'element' >
<!ELEMENT element (XMI.reference) >

<!ENTITY % CommentAssociations '%ModelElementAssociations;,
                                %element;*' >

<!ENTITY % CommentCompositions '%ModelElementCompositions;' >

<!ELEMENT Comment (%remoteContent; |
                          (%CommentProperties;,
                           %CommentAssociations;,
                           %CommentCompositions;)) >
<!ATTLIST Comment
               %XMI.ElementAttributes;
>
<!-- ____________________________________________________________________ -->
<!--                                                                    -->
<!-- UML CLASS: Component                                               -->

<!-- The definition of the Component class is on pages 44, 45, and 48. -->

<!-- UML INCONSISTENCY:  The semantics document shows Component         -->
<!-- inheriting from Classifier on page 44, however the definition on   -->
<!-- page 45 indicates that it inherits from Class.  This DTD defines   -->
<!-- it as inheriting from Class.                                       -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ComponentProperties '%ClassProperties;' >

<!-- ____________________________________________________________ -->
<!--                                                              -->
```

```
<!-- deployment should contain a reference to a Node            -->
<!-- _____________________________________________________ -->

<!ENTITY % deployment 'deployment' >
<!ELEMENT deployment (XMI.reference) >

<!ENTITY % ComponentAssociations '%ClassAssociations;,
                                      %deployment;*' >

<!ENTITY % ComponentCompositions '%ClassCompositions;' >

<!ELEMENT Component (%remoteContent; |
                             (%ComponentProperties;,
                              %ComponentAssociations;,
                              %ComponentCompositions;)) >
<!ATTLIST Component
              %XMI.ElementAttributes;
>
<!-- _____________________________________________________________ -->
<!--                                                                -->
<!-- UML CLASS: Dependency                                          -->

<!-- NOTE: The XML ELEMENT for this class is declared in the core       -->
<!-- package section above.                                             -->
<!-- _____________________________________________________________ -->

<!-- _____________________________________________________________ -->
<!--                                                                -->
<!-- UML CLASS: ModelElement                                        -->

<!-- NOTE: The DTD entities for this class are declared in the core     -->
<!-- package section above.                                             -->
<!-- _____________________________________________________________ -->

<!-- _____________________________________________________________ -->
<!--                                                                -->
<!-- UML CLASS: Node                                                -->

<!-- The definition of the Node class is on pages 44, 46, and 48.       -->

<!-- UML INCONSISTENCY:  The semantics document shows Node inheriting   -->
<!-- from Classifier on page 44, however the definition on page 45      -->
<!-- indicates that it inherits from Class.  This DTD defines it as     -->
<!-- inheriting from Class.                                             -->
<!-- _____________________________________________________________ -->

<!ENTITY % NodeProperties '%ClassProperties;' >

<!ENTITY % component 'component' >
<!ELEMENT component (XMI.reference) >

<!ENTITY % NodeAssociations '%ClassAssociations;,
                                  %component;' >

<!ENTITY % NodeCompositions '%ClassCompositions;' >
```

```
<!ELEMENT Node (%remoteContent; |
                      (%NodeProperties;,
                       %NodeAssociations;,
                       %NodeCompositions;)) >
<!ATTLIST Node
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Presentation                                              -->

<!-- The definition of the Presentation class is on pages 43, 46, and  -->
<!-- 48.                                                                  -->

<!-- UML DEVIATION:  This DTD does not have an element for the          -->
<!-- Presentation class.  Instead, the attributes geometry and style   -->
<!-- are declared in the ViewElement XML ELEMENT.                         -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Refinement                                                -->

<!-- The definition of the Refinement class is on pages 43, 46, 47,     -->
<!-- and 48.                                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % mapping 'mapping' >
<!ELEMENT mapping (#PCDATA) >

<!ENTITY % RefinementProperties '%DependencyProperties;,
                                      %mapping;' >

<!ENTITY % RefinementAssociations '%DependencyAssociations;' >

<!ENTITY % RefinementCompositions '%DependencyCompositions;' >

<!ELEMENT Refinement (%remoteContent; |
                             (%RefinementProperties;,
                              %RefinementAssociations;,
                              %RefinementCompositions;)) >
<!ATTLIST Refinement
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Trace                                                     -->

<!-- The definition of the Trace class is on pages 43, 47, and 49.      -->
<!-- ____________________________________________________________________ -->

<!ENTITY % TraceProperties '%DependencyProperties;' >
```

```
<!ENTITY % TraceAssociations '%DependencyAssociations;' >

<!ENTITY % TraceCompositions '%DependencyCompositions;' >

<!ELEMENT Trace (%remoteContent; |
                        (%TraceProperties;,
                         %TraceAssociations;,
                         %TraceCompositions;)) >
<!ATTLIST Trace
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Usage                                                     -->

<!-- The definition of the Usage class is on pages 43, 47, and 49.      -->
<!-- ____________________________________________________________________ -->

<!ENTITY % UsageProperties '%DependencyProperties;' >

<!ENTITY % UsageAssociations '%DependencyAssociations;' >

<!ENTITY % UsageCompositions '%DependencyCompositions;' >

<!ELEMENT Usage (%remoteContent; |
                        (%UsageProperties;,
                         %UsageAssociations;,
                         %UsageCompositions;)) >
<!ATTLIST Usage
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ViewElement                                               -->

<!-- The definition of the ViewElement class is on pages 43, 47, and    -->
<!-- 49.                                                                  -->

<!-- UML DEVIATION:  Attributes geometry and style are from  UML        -->
<!-- association class Presentation.  Also, tag values are included     -->
<!-- although ViewElement is not a ModelElement in UML.  This was done -->
<!-- to enable model tools to store UI information as tag values.       -->
<!-- In addition, association "model" is represented by element         -->
<!-- "modelElement", and another association, "viewElementReference"    -->
<!-- is included, and this DTD allows ViewElements to contain other     -->
<!-- ViewElements.  All of these deviations were implemented to         -->
<!-- allow modeling tools to store UI information in a more structured -->
<!-- way than defined in UML.                                             -->
<!-- ____________________________________________________________________ -->

<!ELEMENT ViewElement (%ModelElementProperties;,
                           (TaggedValue)*,
```

```
                          (modelElement | ViewElement | viewElement)* ) >
<!ATTLIST ViewElement
              %XMI.ElementAttributes;
>

<!ELEMENT modelElement (XMI.reference) >

<!ELEMENT viewElement (XMI.reference) >

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                            CHAPTER 6                                 -->
<!-- This section contains the declarations of the elements               -->
<!-- corresponding to the extension mechanisms subpackage of the          -->
<!-- foundation package of the UML metamodel.  These classes are DTD      -->
<!-- elements to allow modeling tools to use UML extensions to specify    -->
<!-- other model constructs and save them in this DTD format.             -->
<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- There are no abstract classes in this package.  The  -->
<!-- XML elements corresponding to the concrete classes are -->
<!-- declared in this section.                            -->
<!-- ____________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Constraint                                                -->

<!-- The definition of the Constraint class is on pages 53, 54, and       -->
<!-- 56.                                                                  -->

<!-- NOTE:  The constraint element corresponding to this class is in      -->
<!-- the core package section.                                            -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ModelElement                                              -->

<!-- The definition of the ModelElement class is on pages 53, 54, 56,     -->
<!-- and 57.                                                              -->

<!-- NOTE:  The elementProperties and elementStructure entities           -->
<!-- corresponding to this class are declared in the core package         -->
<!-- section.                                                             -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Stereotype                                                -->
```

```
<!-- The definition of the Stereotype class is on pages 53, 54, 55,     -->
<!-- and 56.                                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % baseClass ‘baseClass’ >
<!ELEMENT baseClass (#PCDATA) >

<!ENTITY % icon ‘icon’ >
<!ELEMENT icon (#PCDATA) >

<!ENTITY % StereotypeProperties ‘%GeneralizableElementProperties;,
                                 %baseClass;,
                                 %icon;’ >

<!ENTITY % stereotypeConstraint ‘stereotypeConstraint’ >
<!ELEMENT stereotypeConstraint (XMI.reference) >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- extendedElement may contain a reference to any ModelElement        -->
<!-- ____________________________________________________________________ -->

<!ENTITY % extendedElement ‘extendedElement’ >
<!ELEMENT extendedElement (XMI.reference) >

<!ENTITY % StereotypeAssociations ‘%GeneralizableElementAssociations;,
                                   (%stereotypeConstraint; |
                                    %extendedElement;)*’ >

<!ENTITY % requiredTag ‘requiredTag’ >
<!ELEMENT requiredTag (TaggedValue) >

<!ENTITY % StereotypeCompositions ‘%GeneralizableElementCompositions;,
                                   %requiredTag;*’ >

<!ELEMENT Stereotype (%remoteContent; |
                      (%StereotypeProperties;,
                       %StereotypeAssociations;,
                       %StereotypeCompositions;)) >
<!ATTLIST Stereotype
          %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: TaggedValue                                               -->

<!-- The definition of the TaggedValue class is on pages 53, 55, 56,    -->
<!-- and 57.                                                              -->
<!-- ____________________________________________________________________ -->

<!ELEMENT TaggedValue (tag,
                       value,
                       constrainedStereotype?) >
```

```
<!ELEMENT tag (#PCDATA) >

<!ELEMENT value (#PCDATA) >

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                            CHAPTER 7                                 -->
<!-- This section contains the declarations of the elements               -->
<!-- corresponding to the data types subpackage of the foundation         -->
<!-- package of the UML metamodel.  The pre-defined enumerations are      -->
<!-- included at the beginning of this DTD, since XML entities must be    -->
<!-- declared before being used.                                          -->
<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: EnumerationLiteral                                        -->

<!-- The definition of the EnumerationLiteral class is on pages 60 and -->
<!-- 61.                                                                  -->
<!-- ____________________________________________________________________ -->

<!ENTITY % EnumerationLiteralProperties '%modelElementName;' >

<!ELEMENT EnumerationLiteral (%EnumerationLiteralProperties;) >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Enumeration                                               -->

<!-- The definition of the Enumeration class is on pages 60 and 61.    -->
<!-- ____________________________________________________________________ -->

<!ENTITY % EnumerationProperties '%DataTypeProperties;' >

<!ENTITY % EnumerationAssociations '%DataTypeAssociations;' >

<!ENTITY % literal 'literal' >
<!ELEMENT literal (EnumerationLiteral) >

<!ENTITY % EnumerationCompositions '%DataTypeCompositions;,
                                    %literal;+' >

<!ELEMENT Enumeration (%remoteContent; |
                       (%EnumerationProperties;,
                        %EnumerationAssociations;,
                        %EnumerationCompositions;)) >
<!ATTLIST Enumeration
          %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
```

```
<!--                                                                    -->
<!--                              CHAPTER 8                             -->
<!-- This section contains the declarations of the elements              -->
<!-- corresponding to the common behavior subpackage of the behavioral  -->
<!-- elements package of the UML metamodel.                             -->
<!-- ___________________________________________________________________ -->
<!-- ___________________________________________________________________ -->

<!-- ___________________________________________________________________ -->
<!--                                                                    -->
<!-- This section contains the declarations of the elements              -->
<!-- corresponding to the abstract classes in the common behavior        -->
<!-- package of the UML metamodel.                                      -->
<!-- ___________________________________________________________________ -->

<!-- ___________________________________________________________________ -->
<!--                                                                    -->
<!-- UML CLASS: Action                                                  -->

<!-- The definition of the Action class is on pages 67, and 68.         -->

<!-- UML INCONSISTENCY:  The attributes of Action on page 68 do not     -->
<!-- include all of the attributes in the diagram on page 67. Also,     -->
<!-- figure 13 on page 67 shows the Action class as concrete.           -->
<!-- ___________________________________________________________________ -->

<!ENTITY % recurrence 'recurrence' >
<!ELEMENT recurrence (#PCDATA) >

<!ENTITY % actionTarget 'actionTarget' >
<!ELEMENT actionTarget (#PCDATA) >

<!ENTITY % isAsynchronous 'isAsynchronous' >
<!ELEMENT isAsynchronous EMPTY >
<!ATTLIST isAsynchronous %Boolean; >

<!ENTITY % script 'script' >
<!ELEMENT script (#PCDATA) >

<!ENTITY % ActionProperties '%ModelElementProperties;,
                             %recurrence;,
                             %actionTarget;,
                             %isAsynchronous;,
                             %script;' >

<!ENTITY % request 'request' >
<!ELEMENT request (XMI.reference) >

<!ENTITY % ActionAssociations '%ModelElementAssociations;,
                               %request;?' >

<!ENTITY % actualArgument 'actualArgument' >
<!ELEMENT actualArgument (Argument) >

<!ENTITY % ActionCompositions '%ModelElementCompositions;,
```

```
                                        %actualArgument;*' >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Instance                                                  -->

<!-- The definition of the Instance class is on pages 67, 70, 74, and  -->
<!-- 75.                                                                  -->

<!-- UML INCONSISTENCY:  Figure 14 on page 67 shows Instance as a       -->
<!-- concrete class.                                                      -->
<!-- ____________________________________________________________________ -->

<!ENTITY % InstanceProperties '%ModelElementProperties;' >

<!ENTITY % linkEnd 'linkEnd' >
<!ELEMENT linkEnd (XMI.reference) >

<!ENTITY % classifier 'classifier' >
<!ELEMENT classifier (XMI.reference) >

<!ENTITY % InstanceAssociations '%ModelElementAssociations;,
                                        %classifier;+,
                                        %linkEnd;*' >

<!ENTITY % instanceSlot 'slot' >
<!ELEMENT slot (AttributeLink) >

<!ENTITY % InstanceCompositions '%ModelElementCompositions;,
                                        %instanceSlot;*' >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Request                                                   -->

<!-- The definition of the Request class is on pages 66, 67, 72, and   -->
<!-- 76.                                                                  -->
<!-- ____________________________________________________________________ -->

<!ENTITY % RequestProperties '%ModelElementProperties;' >

<!ENTITY % RequestAssociations '%ModelElementAssociations;' >

<!ENTITY % RequestCompositions '%ModelElementCompositions;' >

<!ELEMENT Request (%remoteContent; |
                          (%RequestProperties;,
                           %RequestAssociations;,
                           %RequestCompositions;)) >
<!ATTLIST Request
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
```

```
<!-- This section contains the declarations of the elements               -->
<!-- corresponding to the concrete classes in the common behavior        -->
<!-- package of the UML metamodel.                                         -->
<!-- _____________________________________________________________________ -->

<!-- _____________________________________________________________________ -->
<!--                                                                       -->
<!-- UML CLASS: ActionSequence                                             -->

<!-- The definition of the ActionSequence class is on pages 67, and 68 -->
<!-- _____________________________________________________________________ -->

<!ENTITY % ActionSequenceProperties '%ModelElementProperties;' >

<!ENTITY % ActionSequenceAssociations '%ModelElementAssociations;' >

<!ENTITY % action 'action' >
<!ELEMENT action (CreateAction | CallAction | LocalInvocation | ReturnAction |
           SendAction | TerminateAction | DestroyAction | UninterpretedAction)
>

<!ENTITY % ActionSequenceCompositions '%ModelElementCompositions;,
                                       action*' >

<!ELEMENT ActionSequence (%remoteContent; |
                                    (%ActionSequenceProperties;,
                                     %ActionSequenceAssociations;,
                                     %ActionSequenceCompositions;)) >
<!ATTLIST ActionSequence
                %XMI.ElementAttributes;
>

<!-- _____________________________________________________________________ -->
<!--                                                                       -->
<!-- UML CLASS: Argument                                                   -->

<!-- The definition of the Argument class is on pages 67, and 68.       -->
<!-- _____________________________________________________________________ -->

<!ENTITY % argumentValue 'argumentValue' >
<!ELEMENT argumentValue (#PCDATA) >

<!ENTITY % ArgumentProperties '%argumentValue;' >

<!ELEMENT Argument (%remoteContent; | %ArgumentProperties;) >
<!ATTLIST Argument
                 %XMI.ElementAttributes;
>

<!-- _____________________________________________________________________ -->
<!--                                                                       -->
<!-- UML CLASS: AttributeLink                                              -->

<!-- The definition of the AttributeLink class is on pages 67, 69, and -->
<!-- 73.                                                                   -->
```

```
<!-- UML DEVIATION:  The value association is implemented as the        -->
<!-- attributeValue element in this DTD.                                 -->
<!-- ___________________________________________________________________ -->

<!ENTITY % AttributeLinkProperties '%ModelElementProperties;' >

<!ENTITY % attributeValue 'attributeValue' >
<!ELEMENT attributeValue (DataValue) >

<!ENTITY % attribute 'attribute' >
<!ELEMENT attribute (XMI.reference) >

<!ENTITY % AttributeLinkAssociations '%ModelElementAssociations;,
                                      %attribute;' >

<!ENTITY % AttributeLinkCompositions '%ModelElementCompositions;,
                                      %attributeValue;' >

<!ELEMENT AttributeLink (%remoteContent; |
                         (%AttributeLinkProperties;,
                          %AttributeLinkAssociations;,
                          %AttributeLinkCompositions;)) >

<!ATTLIST AttributeLink
          %XMI.ElementAttributes;
>

<!-- ___________________________________________________________________ -->
<!--                                                                     -->
<!-- UML CLASS: CallAction                                               -->

<!-- The definition of the CallAction class is on pages 67, 69, 73,     -->
<!-- and 74.                                                             -->
<!-- ___________________________________________________________________ -->

<!ENTITY % mode 'mode' >
<!ELEMENT mode EMPTY >
<!ATTLIST mode %SynchronousKind; >

<!ENTITY % CallActionProperties '%ActionProperties;,
                                 %mode;' >

<!ENTITY % CallActionAssociations '%ActionAssociations;' >

<!ENTITY % CallActionCompositions '%ActionCompositions;' >

<!ELEMENT CallAction (%remoteContent; |
                      (%CallActionProperties;,
                       %CallActionAssociations;,
                       %CallActionCompositions;)) >
<!ATTLIST CallAction
          %XMI.ElementAttributes;
>
```

```
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: CreateAction                                              -->

<!-- The definition of the CreateAction class is on pages 67, 69, and  -->
<!-- 74.                                                                  -->

<!-- UML INCONSISTENCY:  The UML Semantics document incorrectly calls  -->
<!-- the instantiation association “classifier” on page 69.              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % CreateActionProperties ‘%ActionProperties;’ >

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- instantiation should contain a reference to a Classifier  -->
<!-- ____________________________________________________________ -->

<!ENTITY % instantiation ‘instantiation’ >
<!ELEMENT instantiation (XMI.reference) >

<!ENTITY % CreateActionAssociations ‘%ActionAssociations;,
                                        %instantiation;’ >

<!ENTITY % CreateActionCompositions ‘%ActionCompositions;’ >

<!ELEMENT CreateAction (%remoteContent; |
                                (%CreateActionProperties;,
                                 %CreateActionAssociations;,
                                 %CreateActionCompositions;)) >
<!ATTLIST CreateAction
              %XMI.ElementAttributes;
>
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: DestroyAction                                             -->

<!-- The definition of the DestroyAction class is on pages 67, 69, 70, -->
<!-- and 74.                                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % DestroyActionProperties ‘%ActionProperties;’ >

<!ENTITY % DestroyActionAssociations ‘%ActionAssociations;’ >

<!ENTITY % DestroyActionCompositions ‘%ActionCompositions;’ >

<!ELEMENT DestroyAction (%remoteContent; |
                                 (%DestroyActionProperties;,
                                  %DestroyActionAssociations;,
                                  %DestroyActionCompositions;)) >
<!ATTLIST DestroyAction
              %XMI.ElementAttributes;
>
```

```
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: DataValue                                                 -->

<!-- The definition of the DataValue class is on pages 67, 70, and 74. -->

<!-- UML DEVIATION:  Values are represented as character data in this  -->
<!-- DTD, not as subclasses of the Instance element.                      -->
<!-- ____________________________________________________________________ -->

<!ENTITY % DataValueProperties '%InstanceProperties;' >

<!ENTITY % DataValueAssociations '%InstanceAssociations;' >

<!ENTITY % DataValueCompositions '%InstanceCompositions;' >

<!ELEMENT DataValue (%remoteContent; |
                            (%DataValueProperties;,
                             %DataValueAssociations;,
                             %DataValueCompositions;)) >
<!ATTLIST DataValue
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Signal                                                    -->

<!-- The definition of the Signal class is on pages 66, 73, and 76.     -->
<!-- ____________________________________________________________________ -->

<!ENTITY % SignalProperties '%GeneralizableElementProperties;' >

<!ENTITY % reception 'reception' >
<!ELEMENT reception (XMI.reference) >

<!ENTITY % SignalAssociations '%GeneralizableElementAssociations;,
                                    (%reception; |
                                     %occurrence;)*' >

<!ENTITY % signalParameter 'parameter' >

<!ENTITY % SignalCompositions '%GeneralizableElementCompositions;,
                                    %signalParameter;*' >

<!ELEMENT Signal (%remoteContent; |
                         (%SignalProperties;,
                          %SignalAssociations;,
                          %SignalCompositions;)) >
<!ATTLIST Signal
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
```

```
<!-- UML CLASS: Exception                                                    -->

<!-- The definition of the Exception class is on pages 66, and 70.      -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ExceptionProperties '%SignalProperties;,
                                    %body;' >

<!-- ___________________________________________ -->
<!--                                                 -->
<!-- context should refer to a Collaboration -->
<!-- ___________________________________________ -->

<!ENTITY % context 'context' >
<!ELEMENT context (XMI.reference) >

<!ENTITY % ExceptionAssociations '%SignalAssociations;,
                                      %context;*' >

<!ENTITY % ExceptionCompositions '%SignalCompositions;' >

<!ELEMENT Exception (%remoteContent; |
                            (%ExceptionProperties;,
                             %ExceptionAssociations;,
                             %ExceptionCompositions;)) >
<!ATTLIST Exception
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Link                                                      -->

<!-- The definition of the Link class is on pages 67, 70, 71, and 75.   -->

<!-- UML DEVIATION: Since UML defines a link as containing two          -->
<!-- or more LinkEnds, the linkTarget and linkSource XML ELEMENTS       -->
<!-- represent the two required linkEnds.                                 -->
<!-- ____________________________________________________________________ -->

<!ENTITY % LinkProperties '%ModelElementProperties;' >

<!ENTITY % association 'association' >
<!ELEMENT association (XMI.reference) >

<!ENTITY % LinkAssociations '%ModelElementAssociations;,
                                  %association;' >

<!ENTITY % linkTarget 'linkTarget' >
<!ELEMENT linkTarget (linkEnd) >

<!ENTITY % linkSource 'linkSource' >
<!ELEMENT linkSource (linkEnd) >

<!ENTITY % linkRole 'linkRole' >
```

```
<!ELEMENT linkRole (linkEnd) >

<!ENTITY % LinkCompositions '%ModelElementCompositions;,
                                %linkTarget;,
                                %linkSource;,
                                %linkRole;*' >

<!ELEMENT Link (%remoteContent; |
                       (%LinkProperties;,
                        %LinkAssociations;,
                        %LinkCompositions;)) >
<!ATTLIST Link
              %XMI.ElementAttributes;
>


<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: LinkEnd                                                   -->

<!-- The definition of the LinkEnd class is on pages 67, 71, and 75.   -->
<!-- ____________________________________________________________________ -->

<!ENTITY % LinkEndProperties '%ModelElementProperties;' >

<!ENTITY % instance 'instance' >
<!ELEMENT instance (XMI.reference) >

<!ENTITY % LinkEndAssociations '%ModelElementAssociations;,
                                    %instance;' >

<!ENTITY % LinkEndCompositions '%ModelElementCompositions;,
                                    %associationEnd;' >

<!ELEMENT LinkEnd (%remoteContent; |
                         (%LinkEndProperties;,
                          %LinkEndAssociations;,
                          %LinkEndCompositions;)) >
<!ATTLIST LinkEnd
               %XMI.ElementAttributes;
>


<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Object                                                    -->

<!-- The definition of the Object class is on pages 67, 72, and 76.    -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ObjectProperties '%InstanceProperties;' >

<!ENTITY % ObjectAssociations '%InstanceAssociations;' >

<!ENTITY % ObjectCompositions '%InstanceCompositions;' >
```

```
<!ELEMENT Object (%remoteContent; |
                        (%ObjectProperties;,
                         %ObjectAssociations;,
                         %ObjectCompositions;)) >
<!ATTLIST Object
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: LinkObject                                                -->

<!-- The definition of the LinkObject class is on pages 67, 71, and 75 -->
<!-- ____________________________________________________________________ -->

<!ENTITY % LinkObjectProperties '%ObjectProperties;' >

<!ENTITY % LinkObjectAssociations '%ObjectAssociations;,
                                        %association;' >

<!ENTITY % LinkObjectCompositions '%ObjectCompositions;,
                                        %linkTarget;,
                                        %linkSource;,
                                        %linkRole;*' >

<!ELEMENT LinkObject (%remoteContent; |
                               (%LinkObjectProperties;,
                                %LinkObjectAssociations;,
                                %LinkObjectCompositions;)) >
<!ATTLIST LinkObject
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: LocalInvocation                                           -->

<!-- The definition of the LocalInvocation class is on pages 67 and 71 -->
<!-- ____________________________________________________________________ -->

<!ENTITY % LocalInvocationProperties '%ActionProperties;' >

<!ENTITY % LocalInvocationAssociations '%ActionAssociations;' >

<!ENTITY % LocalInvocationCompositions '%ActionCompositions;' >

<!ELEMENT LocalInvocation (%remoteContent; |
                                     (%LocalInvocationProperties;,
                                      %LocalInvocationAssociations;,
                                      %LocalInvocationCompositions;)) >
<!ATTLIST LocalInvocation
              %XMI.ElementAttributes;
>
```

```
<!-- ____________________________________________________________________ -->
<!--                                                                        -->
<!-- UML CLASS: MessageInstance                                             -->

<!-- The definition of the MessageInstance class is on pages 67, 71,    -->
<!-- and 75.                                                            -->
<!-- ____________________________________________________________________ -->

<!ENTITY % MessageInstanceProperties '%ModelElementProperties;' >

<!-- ___________________________________________ -->
<!--                                             -->
<!-- receiver should refer to a ClassfierRole -->
<!-- ___________________________________________ -->

<!ENTITY % receiver 'receiver' >
<!ELEMENT receiver (XMI.reference) >

<!-- ___________________________________________ -->
<!--                                             -->
<!-- sender should refer to a ClassifierRole -->
<!-- ___________________________________________ -->

<!ENTITY % sender 'sender' >
<!ELEMENT sender (XMI.reference) >

<!ENTITY % MessageInstanceAssociations '%ModelElementAssociations;,
                                        %specification;,
                                        %sender;,
                                        %receiver;,
                                        %argument;*' >

<!ENTITY % MessageInstanceCompositions '%ModelElementCompositions;' >

<!ELEMENT MessageInstance (%remoteContent; |
                           (%MessageInstanceProperties;,
                            %MessageInstanceAssociations;,
                            %MessageInstanceCompositions;)) >
<!ATTLIST MessageInstance
                  %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                        -->
<!-- UML CLASS: Reception                                                   -->

<!-- The definition of the Reception class is on pages 67, 72, and 76. -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ReceptionProperties '%BehavioralFeatureProperties;,
                                %isPolymorphic;,
                                %operationSpecification;' >

<!ENTITY % signal 'signal' >
<!ELEMENT signal (XMI.reference) >
```

```
<!ENTITY % ReceptionAssociations '%BehavioralFeatureAssociations;,
                                        %signal;' >

<!ENTITY % ReceptionCompositions '%BehavioralFeatureCompositions;' >

<!ELEMENT Reception (%remoteContent; |
                           (%ReceptionProperties;,
                            %ReceptionAssociations;,
                            %ReceptionCompositions;)) >
<!ATTLIST Reception
                %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ReturnAction                                              -->

<!-- The definition of the ReturnAction class is on pages 67 and 72.    -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ReturnActionProperties '%ActionProperties;' >

<!ENTITY % ReturnActionAssociations '%ActionAssociations;' >

<!ENTITY % ReturnActionCompositions '%ActionCompositions;' >

<!ELEMENT ReturnAction (%remoteContent; |
                                 (%ReturnActionProperties;,
                                  %ReturnActionAssociations;,
                                  %ReturnActionCompositions;)) >
<!ATTLIST ReturnAction
             %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: SendAction                                                -->

<!-- The definition of the SendAction class is on pages 67, 72, 73,     -->
<!-- and 76.                                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % SendActionProperties '%ActionProperties;' >

<!ENTITY % SendActionAssociations '%ActionAssociations;' >

<!ENTITY % SendActionCompositions '%ActionCompositions;' >

<!ELEMENT SendAction (%remoteContent; |
                           (%SendActionProperties;,
                            %SendActionAssociations;,
                            %SendActionCompositions;)) >
<!ATTLIST SendAction
             %XMI.ElementAttributes;
```

```
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: TerminateAction                                           -->

<!-- The definition of the TerminateAction class is on pages 67, 73,    -->
<!-- and 76.                                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % TerminateActionProperties '%ActionProperties;' >

<!ENTITY % TerminateActionAssociations '%ActionAssociations;' >

<!ENTITY % TerminateActionCompositions '%ActionCompositions;' >

<!ELEMENT TerminateAction (%remoteContent; |
                                    (%TerminateActionProperties;,
                                     %TerminateActionAssociations;,
                                     %TerminateActionCompositions;)) >
<!ATTLIST TerminateAction
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: UninterpretedAction                                       -->

<!-- The definition of the UninterpretedAction class is on pages 67     -->
<!-- and 73.                                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % UninterpretedActionProperties '%ActionProperties;,
                                          %body;' >

<!ENTITY % UninterpretedActionAssociations '%ActionAssociations;' >

<!ENTITY % UninterpretedActionCompositions '%ActionCompositions;' >

<!ELEMENT UninterpretedAction (%remoteContent; |
                                       (%UninterpretedActionProperties;,
                                        %UninterpretedActionAssociations;,
                                        %UninterpretedActionCompositions;)) >
<!ATTLIST UninterpretedAction
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                          CHAPTER 9                                   -->
<!-- This section contains the declarations of the elements              -->
<!-- corresponding to the collaborations subpackage of the behavioral    -->
<!-- elements package of the UML metamodel.                               -->
<!-- ____________________________________________________________________ -->
```

```
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- There are no abstract classes defined in the UML metamodel in        -->
<!-- Chapter 9, only concrete classes.                                    -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: AssociationEndRole                                        -->

<!-- The definition of the AssociationEndRole class is on pages 81,     -->
<!-- 82, and 84.                                                          -->
<!-- ____________________________________________________________________ -->

<!ENTITY % AssociationEndRoleProperties '%AssociationEndProperties;' >

<!-- ____________________________________________________________ -->
<!--                                                                -->
<!-- base is used in AssociationRole and ClassifierRole also. -->
<!-- ____________________________________________________________ -->

<!ENTITY % base 'base' >
<!ELEMENT base (XMI.reference) >

<!ENTITY % AssociationEndRoleAssociations '%AssociationEndAssociations;,
                                           %base;' >

<!ENTITY % AssociationEndRoleCompositions '%AssociationEndCompositions;' >

<!ELEMENT AssociationEndRole (%remoteContent; |
                              (%AssociationEndRoleProperties;,
                               %AssociationEndRoleAssociations;,
                               %AssociationEndRoleCompositions;)) >
<!ATTLIST AssociationEndRole
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: AssociationRole                                           -->

<!-- The definition of the AssociationRole class is on pages 81, 82,    -->
<!-- and 84.                                                              -->

<!-- NOTE:  The connection association in the diagram on page 81 does   -->
<!-- not appear in the description of this class on page 82.              -->

<!-- UML DEVIATION:  The connection association is implemented by       -->
<!-- XML elements sourceEndRole, targetEndRole, and connectionEndRole. -->
<!-- ____________________________________________________________________ -->

<!ENTITY % AssociationRoleProperties '%AssociationProperties;,
                                      %multiplicity;' >
```

```
<!ENTITY % AssociationRoleAssociations '%AssociationAssociations;,
                                        %base;' >

<!ENTITY % sourceEndRole 'sourceEndRole' >
<!ELEMENT sourceEndRole (associationEndRole) >

<!ENTITY % targetEndRole 'targetEndRole' >
<!ELEMENT targetEndRole (associationEndRole) >

<!ENTITY % connectionEndRole 'connectionEndRole' >
<!ELEMENT connectionEndRole (associationEndRole) >

<!ENTITY % AssociationRoleCompositions '%AssociationCompositions;,
                                        %sourceEndRole;,
                                        %targetEndRole;,
                                        %connectionEndRole;*' >

<!ELEMENT AssociationRole (%remoteContent; |
                           (%AssociationRoleProperties;,
                            %AssociationRoleAssociations;,
                            %AssociationRoleCompositions;)) >
<!ATTLIST AssociationRole
          %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ClassifierRole                                            -->

<!-- The definition of the ClassifierRole class is on pages 81, 82,     -->
<!-- and 84.                                                              -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ClassifierRoleProperties '%ModelElementProperties;,
                                     %multiplicity;' >

<!ENTITY % ClassifierRoleAssociations '%ModelElementAssociations;,
                                       %base;' >

<!ENTITY % availableFeature 'availableFeature' >
<!ELEMENT availableFeature (Attribute |
                            Operation |
                            Method |
                            Reception) >

<!ENTITY % ClassifierRoleCompositions '%ModelElementCompositions;,
                                       %availableFeature;*' >

<!ELEMENT ClassifierRole (%remoteContent; |
                          (%ClassifierRoleProperties;,
                           %ClassifierRoleAssociations;,
                           %ClassifierRoleCompositions;)) >
<!ATTLIST ClassifierRole
          %XMI.ElementAttributes;
```

```
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Collaboration                                             -->

<!-- The definition of the Collaboration class is on pages 81, 82, 83, -->
<!-- 84, and 85.                                                          -->

<!-- UML DEVIATION:  The ownedElement association is implemented by     -->
<!-- explicitly listing classifierRole and associationRole in the       -->
<!-- content model for XML element collaboration.                          -->
<!-- ____________________________________________________________________ -->

<!ENTITY % CollaborationProperties '%ModelElementProperties;' >

<!-- ____________________________________________________ -->
<!--                                                              -->
<!-- constrainingElement can refer to any ModelElement -->
<!-- ____________________________________________________ -->

<!ENTITY % constrainingElement 'constrainingElement' >
<!ELEMENT constrainingElement (XMI.reference) >

<!ENTITY % representedClassifier 'representedClassifier' >
<!ELEMENT representedClassifier (XMI.reference) >

<!ENTITY % representedOperation 'representedOperation' >
<!ELEMENT representedOperation (XMI.reference) >

<!ENTITY % CollaborationAssociations '%ModelElementAssociations;,
                                      %constrainingElement;*,
                                      (%representedClassifier; |
                                       %representedOperation;)' >

<!ENTITY % collaborationInteraction 'interaction' >
<!ELEMENT interaction (Interaction) >

<!ENTITY % CollaborationCompositions '%ModelElementCompositions;,
                                      %collaborationInteraction;*' >

<!ELEMENT Collaboration (%remoteContent; |
                                  (%CollaborationProperties;,
                                   %CollaborationAssociations;,
                                   %CollaborationCompositions;)) >
<!ATTLIST Collaboration
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Interaction                                               -->

<!-- The definition of the Interaction class is on pages 81 and 83.     -->
<!-- ____________________________________________________________________ -->
```

```
<!ENTITY % InteractionProperties '%ModelElementProperties;' >

<!ENTITY % message 'message' >
<!ELEMENT message (XMI.reference) >

<!ENTITY % InteractionAssociations '%ModelElementAssociations;,
                                        %context;,
                                        %message;+' >

<!ENTITY % InteractionCompositions '%ModelElementCompositions;' >

<!ELEMENT Interaction (%remoteContent; |
                                   (%InteractionProperties;,
                                    %InteractionAssociations;,
                                    %InteractionCompositions;)) >
<!ATTLIST Interaction
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Message                                                   -->

<!-- The definition of the Message class is on pages 81, 83, 84, and    -->
<!-- 85.                                                                  -->

<!-- UML DEVIATION:  The action association on page 81 is implemented  -->
<!-- by XML element messageAction, since element action is used to      -->
<!-- define element actionSequence.                                       -->
<!-- ____________________________________________________________________ -->

<!ENTITY % MessageProperties '%ModelElementProperties;' >

<!-- _____________________________________ -->
<!--                                       -->
<!-- activator should refer to a Message -->
<!-- _____________________________________ -->

<!ENTITY % activator 'activator' >
<!ELEMENT activator (XMI.reference) >

<!ENTITY % messageAction 'messageAction' >
<!ELEMENT messageAction (XMI.reference) >

<!-- _____________________________________ -->
<!--                                         -->
<!-- predecessor should refer to a Message -->
<!-- _____________________________________ -->

<!ENTITY % predecessor 'predecessor' >
<!ELEMENT predecessor (XMI.reference) >

<!ENTITY % MessageAssociations '%ModelElementAssociations;,
                                     %activator;,
```

```
                                        %base;,
                                        %receiver;,
                                        %predecessor;*,
                                        %sender;,
                                        %messageAction;' >

<!ENTITY % MessageCompositions '%ModelElementCompositions;' >

<!ELEMENT Message (%remoteContent; |
                           (%MessageProperties;,
                            %MessageAssociations;,
                            %MessageCompositions;)) >
<!ATTLIST Message
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                              CHAPTER 10                              -->
<!-- This section contains the declarations of the elements               -->
<!-- corresponding to the use case subpackage of the behavioral           -->
<!-- elements package of the UML metamodel.                               -->
<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Actor                                                     -->

<!-- The definition of the Actor class is on page 90.                     -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ActorProperties '%ClassifierProperties;' >

<!ENTITY % ActorAssociations '%ClassifierAssociations;' >

<!ENTITY % ActorCompositions '%ClassifierCompositions;' >

<!ELEMENT Actor (%remoteContent; |
                         (%ActorProperties;,
                          %ActorAssociations;,
                          %ActorCompositions;)) >
<!ATTLIST Actor
               %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: UseCase                                                   -->

<!-- The definition of the UseCase class is on pages 90, 91, and 92.    -->

<!-- UML DEVIATION:  The extensionPoint attribute is implemented by     -->
<!-- the extensionPoint XML element since it is a list of Strings.      -->
```

```
<!-- ____________________________________________________________________ -->

<!ENTITY % UsecaseProperties '%ClassifierProperties;' >

<!ENTITY % UsecaseAssociations '%ClassifierAssociations;' >

<!ENTITY % extensionPoint 'extensionPoint' >
<!ELEMENT extensionPoint (#PCDATA) >

<!ENTITY % UsecaseCompositions '%ClassifierCompositions;,
                                ViewElement*,
                                %extensionPoint;*' >

<!ELEMENT Usecase (%remoteContent; |
                   (%UsecaseProperties;,
                    %UsecaseAssociations;,
                    %UsecaseCompositions;)) >
<!ATTLIST Usecase
          %XMI.ElementAttributes;
>


<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: UseCaseInstance                                           -->

<!-- The definition of the UseCaseInstance class is on pages 90 and 91 -->
<!-- ____________________________________________________________________ -->

<!ENTITY % UsecaseInstanceProperties '%InstanceProperties;' >

<!ENTITY % UsecaseInstanceAssociations '%InstanceAssociations;' >

<!ENTITY % UsecaseInstanceCompositions '%InstanceCompositions;' >

<!ELEMENT UsecaseInstance (%remoteContent; |
                           (%UsecaseInstanceProperties;,
                            %UsecaseInstanceAssociations;,
                            %UsecaseInstanceCompositions;)) >
<!ATTLIST UsecaseInstance
          %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                          CHAPTER 11                                  -->
<!-- This section contains the declarations of the elements               -->
<!-- corresponding to the state machine subpackage of the behavioral      -->
<!-- elements package of the UML metamodel.                               -->
<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________ -->
<!--                                                              -->
```

```
<!-- This section contains the XML entities corresponding to the -->
<!-- abstract classes in the state machine package.                -->
<!-- _____________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Event                                                     -->

<!-- The definition of the Event class is on pages 98 and 100. There    -->
<!-- is no element in this DTD for the Event class in the UML            -->
<!-- UML metamodel because it is an abstract class that does not         -->
<!-- define new associations or attributes.                               -->
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: StateVertex                                               -->

<!-- The definition of the StateVertex class is on pages 98 and 102.    -->
<!-- ____________________________________________________________________ -->

<!ENTITY % StateVertexProperties '%ModelElementProperties;' >

<!-- _____________________________________________________ -->
<!--                                                       -->
<!-- parent should refer to the enclosing CompositeState   -->
<!-- _____________________________________________________ -->

<!ENTITY % parent 'parent' >
<!ELEMENT parent (XMI.reference) >

<!-- _____________________________________________________ -->
<!--                                                       -->
<!-- outgoing and incoming should refer to Transitions     -->
<!-- _____________________________________________________ -->

<!ENTITY % outgoing 'outgoing' >
<!ELEMENT outgoing (XMI.reference) >

<!ENTITY % incoming 'incoming' >
<!ELEMENT incoming (XMI.reference) >

<!ENTITY % StateVertexAssociations '%ModelElementAssociations;,
                                    %parent;?,
                                    (%outgoing; |
                                     %incoming;)*' >

<!ENTITY % StateVertexCompositions '%ModelElementCompositions;' >

<!-- _____________________________________________________________ -->
<!--                                                               -->
<!-- This section contains the XML elements for the concrete -->
<!-- classes in the state machines package.                  -->
<!-- _____________________________________________________________ -->
```

```
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: CallEvent                                                 -->

<!-- The definition of the CallEvent class is on pages 98 and 99.       -->
<!-- ____________________________________________________________________ -->

<!ENTITY % CallEventProperties '%ModelElementProperties;' >

<!ENTITY % operation 'operation' >
<!ELEMENT operation (XMI.reference) >

<!ENTITY % CallEventAssociations '%ModelElementAssociations;,
                                        %operation;' >

<!ENTITY % CallEventCompositions '%ModelElementCompositions;' >

<!ELEMENT CallEvent (%remoteContent; |
                               (%CallEventProperties;,
                                %CallEventAssociations;,
                                %CallEventCompositions;)) >
<!ATTLIST CallEvent
              %XMI.ElementAttributes;
>


<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ChangeEvent                                               -->

<!-- The definition of the ChangeEvent class is on pages 98 and 99.     -->
<!-- ____________________________________________________________________ -->

<!ENTITY % changeExpression 'changeExpression' >
<!ELEMENT changeExpression (#PCDATA) >

<!ENTITY % ChangeEventProperties '%ModelElementProperties;,
                                       %changeExpression;' >

<!ENTITY % ChangeEventAssociations '%ModelElementAssociations;' >

<!ENTITY % ChangeEventCompositions '%ModelElementCompositions;' >

<!ELEMENT ChangeEvent (%remoteContent; |
                                  (%ChangeEventProperties;,
                                   %ChangeEventAssociations;,
                                   %ChangeEventCompositions;)) >
<!ATTLIST ChangeEvent
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: State                                                     -->
```

```
<!-- The definition of the State class is on pages 98 and 101.          -->

<!-- UML DEVIATION:  The internalTransition association is implemented -->
<!-- by including the transition element in the stateStructure entity. -->
<!-- ____________________________________________________________________ -->

<!ENTITY % StateProperties '%StateVertexProperties;' >

<!ENTITY % deferredEvent 'deferredEvent' >
<!ELEMENT deferredEvent (XMI.reference) >

<!ENTITY % StateAssociations '%StateVertexAssociations;,
                                    %deferredEvent;*' >

<!ENTITY % entry 'entry' >
<!ELEMENT entry (ActionSequence) >

<!ENTITY % exit 'exit' >
<!ELEMENT exit (ActionSequence) >

<!ENTITY % internalTransition 'internalTransition' >
<!ELEMENT internalTransition (Transition) >

<!ENTITY % StateCompositions '%StateVertexCompositions;,
                                    %entry;?,
                                    %exit;?,
                                    %internalTransition;*' >

<!ELEMENT State (%remoteContent; |
                        (%StateProperties;,
                         %StateAssociations;,
                         %StateCompositions;)) >
<!ATTLIST State
                %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: CompositeState                                            -->

<!-- The definition of the CompositeState class is on pages 98, 99,     -->
<!-- 100, 103, and 104.                                                   -->

<!-- UML DEVIATION:  The derived attribute, isRegion, is not defined    -->
<!-- in this DTD.  The association substates is implemented by          -->
<!-- explicitly indicating the types of states in the content model of -->
<!-- the compositeState XML element.                                      -->
<!-- ____________________________________________________________________ -->

<!ENTITY % isConcurrent 'isConcurrent' >
<!ELEMENT isConcurrent EMPTY >
<!ATTLIST isConcurrent %Boolean; >

<!ENTITY % CompositeStateProperties '%StateProperties;,
                                        %isConcurrent;' >
```

```
<!ENTITY % CompositeStateAssociations '%StateAssociations;' >

<!ENTITY % substate 'substate' >
<!ELEMENT substate (PseudoState |
                        SimpleState |
                        SubmachineState |
                        ActivityState |
                        ActionState |
                        ObjectFlowState) >

<!ENTITY % CompositeStateCompositions '%StateCompositions;,
                                            %substate;+' >

<!ELEMENT CompositeState (%remoteContent; |
                                    (%CompositeStateProperties;,
                                     %CompositeStateAssociations;,
                                     %CompositeStateCompositions;)) >
<!ATTLIST CompositeState
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Guard                                                     -->

<!-- The definition of the Guard class is on pages 98, 100, and 104.    -->
<!-- ____________________________________________________________________ -->

<!ENTITY % expression 'expression' >
<!ELEMENT expression (#PCDATA) >

<!ENTITY % GuardProperties '%ModelElementProperties;,
                                %expression;' >

<!ENTITY % GuardAssociations '%ModelElementAssociations;' >

<!ENTITY % GuardCompositions '%ModelElementCompositions;' >

<!ELEMENT Guard (%remoteContent; |
                        (%GuardProperties;,
                         %GuardAssociations;,
                         %GuardCompositions;)) >
<!ATTLIST Guard
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: PseudoState                                               -->

<!-- The definition of the PseudoState class is on pages 98, 100, and  -->
<!-- 104.                                                                 -->
<!-- ____________________________________________________________________ -->
```

```
<!ENTITY % stateKind 'stateKind'>
<!ELEMENT stateKind EMPTY >
<!ATTLIST stateKind %PseudoStateKind; >

<!ENTITY % PseudoStateProperties '%StateVertexProperties;,
                                    %stateKind;' >

<!ENTITY % PseudoStateAssociations '%StateVertexAssociations;' >

<!ENTITY % PseudoStateCompositions '%StateVertexCompositions;' >

<!ELEMENT PseudoState (%remoteContent; |
                              (%PseudoStateProperties;,
                               %PseudoStateAssociations;,
                               %PseudoStateCompositions;)) >
<!ATTLIST PseudoState
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: SignalEvent                                               -->

<!-- The definition of the SignalEvent class is on pages 98 and 101.   -->
<!-- 104.                                                                 -->

<!-- NOTE:  The signal element is declared with the reception element. -->
<!-- ____________________________________________________________________ -->

<!ENTITY % SignalEventProperties '%ModelElementProperties;' >

<!ENTITY % SignalEventAssociations '%ModelElementAssociations;,
                                      %signal;' >

<!ENTITY % SignalEventCompositions '%ModelElementCompositions;' >

<!ELEMENT SignalEvent (%remoteContent; |
                            (%SignalEventProperties;,
                             %SignalEventAssociations;,
                             %SignalEventCompositions;)) >
<!ATTLIST SignalEvent
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: SimpleState                                               -->

<!-- The definition of the SimpleState class is on pages 98 and 101.   -->
<!-- ____________________________________________________________________ -->

<!ENTITY % SimpleStateProperties '%StateProperties;' >

<!ENTITY % SimpleStateAssociations '%StateAssociations;' >
```

```
<!ENTITY % SimpleStateCompositions '%StateCompositions;' >

<!ELEMENT SimpleState (%remoteContent; |
                              (%SimpleStateProperties;,
                               %SimpleStateAssociations;,
                               %SimpleStateCompositions;)) >
<!ATTLIST SimpleState
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: StateMachine                                              -->

<!-- The definition of the StateMachine class is on pages 98, 101,      -->
<!-- 104, and 105.                                                        -->

<!-- NOTE:  The element context is declared with element interaction.   -->

<!-- UML DEVIATION:  The association transitions is implemented by       -->
<!-- including the element transition in the content model of element   -->
<!-- StateMachine.                                                        -->
<!-- ____________________________________________________________________ -->

<!ENTITY % StateMachineProperties '%ModelElementProperties;' >

<!ENTITY % StateMachineAssociations '%ModelElementAssociations;,
                                          %context;' >

<!ENTITY % top 'top' >
<!ELEMENT top (CompositeState) >

<!ENTITY % transitions 'transitions' >
<!ELEMENT transitions (Transition)* >

<!ENTITY % StateMachineCompositions '%ModelElementCompositions;,
                                          %top;,
                                          %transitions;?' >

<!ELEMENT StateMachine (%remoteContent; |
                                  (%StateMachineProperties;,
                                   %StateMachineAssociations;,
                                   %StateMachineCompositions;)) >
<!ATTLIST StateMachine
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: SubmachineState                                           -->

<!-- The definition of the SubMachineState class is on pages 98, 102,   -->
<!-- and 103.                                                             -->
<!-- ____________________________________________________________________ -->
```

```
<!ENTITY % SubmachineStateProperties '%StateProperties;' >

<!-- ____________________________________________________________ -->
<!--                                                                -->
<!-- submachine should contain a reference to a StateMachine -->
<!-- ____________________________________________________________ -->

<!ENTITY % submachine 'submachine' >
<!ELEMENT submachine (XMI.reference) >

<!ENTITY % SubmachineStateAssociations '%StateAssociations;,
                                        %submachine;' >

<!ENTITY % SubmachineStateCompositions '%StateCompositions;' >

<!ELEMENT SubmachineState (%remoteContent; |
                           (%SubmachineStateProperties;,
                            %SubmachineStateAssociations;,
                            %SubmachineStateCompositions;)) >
<!ATTLIST SubmachineState
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: TimeEvent                                                 -->

<!-- The definition of the TimeEvent class is on pages 98 and 103.      -->
<!-- ____________________________________________________________________ -->

<!ENTITY % duration 'duration'>
<!ELEMENT duration (#PCDATA) >

<!ENTITY % TimeEventProperties '%ModelElementProperties;,
                                %duration;' >

<!ENTITY % TimeEventAssociations '%ModelElementAssociations;' >

<!ENTITY % TimeEventCompositions '%ModelElementCompositions;' >

<!ELEMENT TimeEvent (%remoteContent; |
                     (%TimeEventProperties;,
                      %TimeEventAssociations;,
                      %TimeEventCompositions;)) >
<!ATTLIST TimeEvent
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Transition                                                -->

<!-- The definition of the Transition class is on pages 98, 103, 105,   -->
<!-- and 106.                                                             -->
```

```
<!-- UML DEVIATION:  The source and target associations are              -->
<!-- implemented with sourceState and targetState elements because      -->
<!-- elements source and target are used elsewhere in this DTD.         -->
<!-- ___________________________________________________________________ -->

<!ENTITY % TransitionProperties '%ModelElementProperties;' >

<!ENTITY % sourceState 'sourceState' >
<!ELEMENT sourceState (XMI.reference) >

<!ENTITY % targetState 'targetState' >
<!ELEMENT targetState (XMI.reference) >

<!ENTITY % TransitionAssociations '%ModelElementAssociations;,
                                   %sourceState;,
                                   %targetState;' >

<!ENTITY % guard 'guard' >
<!ELEMENT guard (Guard) >

<!ENTITY % effect 'effect' >
<!ELEMENT effect (ActionSequence) >

<!ENTITY % trigger 'trigger' >
<!ELEMENT trigger (SignalEvent | CallEvent | TimeEvent | ChangeEvent) >

<!ENTITY % TransitionCompositions '%ModelElementCompositions;,
                                   %trigger;?,
                                   %guard;?,
                                   %effect;?' >

<!ELEMENT Transition (%remoteContent; |
                              (%TransitionProperties;,
                               %TransitionAssociations;,
                               %TransitionCompositions;)) >
<!ATTLIST Transition
              %XMI.ElementAttributes;
>

<!-- ___________________________________________________________________ -->
<!-- ___________________________________________________________________ -->
<!--                                                                     -->
<!--                          CHAPTER 11                                 -->
<!-- This section contains the declarations of the elements              -->
<!-- corresponding to the activity model subpackage of the behavioral    -->
<!-- elements package of the UML metamodel.                              -->
<!-- ___________________________________________________________________ -->
<!-- ___________________________________________________________________ -->

<!-- ___________________________________________________________________ -->
<!--                                                                     -->
<!-- UML CLASS: ActivityModel                                            -->

<!-- The definition of the ActivityModel class is on pages 121, 122,    -->
<!-- and 124.                                                            -->
```

```
<!-- ____________________________________________________________________ -->

<!ENTITY % ActivityModelProperties '%StateMachineProperties;' >

<!ENTITY % ActivityModelAssociations '%StateMachineAssociations;' >

<!ENTITY % partition 'partition' >
<!ELEMENT partition (Partition) >

<!ENTITY % ActivityModelCompositions '%StateMachineCompositions;,
                                      partition*' >

<!ELEMENT ActivityModel (%remoteContent; |
                         (%ActivityModelProperties;,
                          %ActivityModelAssociations;,
                          %ActivityModelCompositions;)) >
<!ATTLIST ActivityModel
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ActionState                                               -->

<!-- The definition of the ActionState class is on pages 121, 122,      -->
<!-- and 124.                                                             -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ActionStateProperties '%SimpleStateProperties;' >

<!ENTITY % ActionStateAssociations '%SimpleStateAssociations;' >

<!ENTITY % ActionStateCompositions '%SimpleStateCompositions;' >

<!ELEMENT ActionState (%remoteContent; |
                       (%ActionStateProperties;,
                        %ActionStateAssociations;,
                        %ActionStateCompositions;)) >
<!ATTLIST ActionState
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ActivityState                                             -->

<!-- The definition of the ActivityState class is on pages 121 and 122 -->

<!-- UML INCONSISTENCY:  The diagram on page 121 shows ActivityState   -->
<!-- inheriting from SimpleState, but the description on page 122       -->
<!-- indicates that it inherits from SubmachineState.                     -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ActivityStateProperties '%SubmachineStateProperties;' >
```

```
<!ENTITY % ActivityStateAssociations '%SubmachineStateAssociations;' >

<!ENTITY % ActivityStateCompositions '%SubmachineStateCompositions;' >

<!ELEMENT ActivityState (%remoteContent; |
                                  (%ActivityStateProperties;,
                                   %ActivityStateAssociations;,
                                   %ActivityStateCompositions;)) >
<!ATTLIST ActivityState
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ClassifierInState                                         -->

<!-- The definition of the ClassifierState class is on pages 121 and   -->
<!-- 123.                                                                 -->

<!-- NOTE:  Element type is declared with Attribute.                    -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ClassifierInStateProperties '%ClassifierProperties;' >

<!ENTITY % inState 'inState' >
<!ELEMENT inState (XMI.reference) >

<!ENTITY % ClassifierInStateAssociations '%ClassifierAssociations;,
                                          %structuralFeatureType;,
                                          %inState;' >

<!ENTITY % ClassifierInStateCompositions '%ClassifierCompositions;' >

<!ELEMENT ClassifierInState (%remoteContent; |
                                     (%ClassifierInStateProperties;,
                                      %ClassifierInStateAssociations;,
                                      %ClassifierInStateCompositions;)) >
<!ATTLIST ClassifierInState
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ObjectFlowState                                           -->

<!-- The definition of the ObjectFlowState class is on pages 121, 123, -->
<!-- and 124.                                                             -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ObjectFlowStateProperties '%SimpleStateProperties;' >

<!-- ____________________________________________________________ -->
<!--                                                              -->
<!-- typeState should contain a reference to a ClassifierInState  -->
<!-- ____________________________________________________________ -->
```

```
<!ENTITY % typeState ‘typeState’ >
<!ELEMENT typeState (XMI.reference) >

<!ENTITY % ObjectFlowStateAssociations ‘%SimpleStateAssociations;,
                                          %typeState;’ >

<!ENTITY % ObjectFlowStateCompositions ‘%SimpleStateCompositions;’ >

<!ELEMENT ObjectFlowState (%remoteContent; |
                                    (%ObjectFlowStateProperties;,
                                     %ObjectFlowStateAssociations;,
                                     %ObjectFlowStateCompositions;)) >
<!ATTLIST ObjectFlowState
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Partition                                                 -->

<!-- The definition of the Partition class is on pages 121 and 123.     -->
<!-- ____________________________________________________________________ -->

<!ENTITY % PartitionProperties ‘%ModelElementProperties;’ >

<!-- _____________________________________________________ -->
<!--                                                       -->
<!-- contents should contain a reference to a ModelElement  -->
<!-- _____________________________________________________ -->

<!ENTITY % contents ‘contents’ >
<!ELEMENT contents (XMI.reference) >

<!ENTITY % PartitionAssociations ‘%ModelElementAssociations;,
                                      %contents;*’ >

<!ENTITY % PartitionCompositions ‘%ModelElementCompositions;’ >


<!ELEMENT Partition (%remoteContent; |
                                (%PartitionProperties;,
                                 %PartitionAssociations;,
                                 %PartitionCompositions;)) >
<!ATTLIST Partition
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                          CHAPTER 12                                  -->
<!-- This section contains the declarations of the elements              -->
<!-- corresponding to the model management package of the UML metamodel.-->
<!-- ____________________________________________________________________ -->
```

```
<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: ElementReference                                          -->

<!-- The definition of the ElementReference class is on page 129.       -->
<!-- ____________________________________________________________________ -->

<!ENTITY % alias 'alias' >
<!ELEMENT alias (#PCDATA) >

<!ENTITY % ElementReferenceProperties '%modelElementVisibility;,
                                       %alias;' >

<!ENTITY % ElementReferenceCompositions 'XMI.reference' >

<!ELEMENT ElementReference (%ElementReferenceProperties;,
                            %ElementReferenceCompositions;) >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Package                                                   -->

<!-- The definition of the Package class is on pages 129 - 133.         -->

<!-- NOTE:  The fact that packages can contain activityModels is        -->
<!-- documented on page 124.                                              -->

<!-- UML DEVIATION:  The UMLVersion and DTDVersion attributes are not  -->
<!-- defined in UML.  They are included to allow tools to recognize     -->
<!-- which version of UML and which version of the DTD are being used. -->
<!-- ____________________________________________________________________ -->

<!ENTITY % PackageProperties '%GeneralizableElementProperties;' >

<!ENTITY % PackageAssociations '%GeneralizableElementAssociations;' >

<!ENTITY % PackageCompositions '%GeneralizableElementCompositions;' >

<!ELEMENT Package (%remoteContent; |
                          (%PackageProperties;,
                           %PackageAssociations;,
                           %PackageCompositions;)) >
<!ATTLIST Package
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Model                                                     -->

<!-- The definition of the Model class is on pages 129 and 130.         -->

<!-- UML DEVIATION:  The UMLVersion and DTDVersion attributes are not  -->
```

```
<!-- defined in UML.  They are included to allow tools to recognize     -->
<!-- which version of UML and which version of the DTD are being used. -->
<!-- ____________________________________________________________________ -->

<!ENTITY % ModelProperties '%PackageProperties;' >

<!ENTITY % ModelAssociations '%PackageAssociations;' >

<!ENTITY % ModelCompositions '%PackageCompositions;' >

<!ELEMENT Model (%remoteContent; |
                        (%ModelProperties;,
                         %ModelAssociations;,
                         %ModelCompositions;)) >
<!ATTLIST Model
              %XMI.ElementAttributes;
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- UML CLASS: Subsystem                                                 -->

<!-- The definition of the Subsystem class is on pages 130, 131, 133,   -->
<!-- and 134.                                                             -->
<!-- ____________________________________________________________________ -->

<!ENTITY % isInstantiable 'isInstantiable' >
<!ELEMENT isInstantiable EMPTY >
<!ATTLIST isInstantiable %Boolean; >

<!ENTITY % SubsystemProperties '%PackageProperties;,
                                    %isInstantiable;' >

<!ENTITY % SubsystemAssociations '%PackageAssociations;,
                                       (%participant; |
                                        %realization; |
                                        %specification;)*' >

<!ENTITY % SubsystemCompositions '%ClassifierCompositions;' >

<!ELEMENT Subsystem (%remoteContent; |
                               (%SubsystemProperties;,
                                %SubsystemAssociations;,
                                %SubsystemCompositions;)) >
<!ATTLIST Subsystem
              %XMI.ElementAttributes;
>
```

MOF 1.1 DTD B

B.1 Introduction

Appendix B contains a DTD generated by hand that represents the MOF model which is described in Section 3 MOF Model and Interfaces of the MOF 1.1 specification . The MOF model is used for describing OMG compliant metamodels, but for the purposes of transfering metamodels, the MOF model itself is treated as a metamodeland therefore has a corresponding DTD. This DTD generally follows the specification of the above section on representing metamodel information. By examining this DTD, you can gain a better understanding of the types of metamodel information that can be represented in an XML DTD, and the information that cannot be specified.

The structure of the DTD closely corresponds to the document "MOF 1.1 Specification, 1 September 1997". Each XML element corresponding to a MOF class has a comment indicating which section and pages of that document describe the class. You can verify the accuracy of the DTD against the document by reading the pages of the document in the comments and verifying that the encoding for them is correct.

B.2 MOF DTD

```
<?xml version="1.0" encoding="utf-8" ?>

<!-- THIS PRODUCT CONTAINS RESTRICTED MATERIALS OF IBM 5639-D57,             -->
<!-- (C) COPYRIGHT International Business Machines Corp., 1998               -->
<!-- All Rights Reserved * Licensed Materials - Property of IBM              -->
<!-- The source code for this program is not published or otherwise divested -->
<!-- of its trade secrets, irrespective of what has been deposited with the  -->
<!-- U. S. Copyright Office.                                                  -
->

<!-- MOF Model Document Type Declaration -->
<!-- Tim Grose & Steve Brodsky            -->
```

```
<!-- Version: 1.1                                -->
<!-- Date: 7/4/98                                -->

<!-- ____________________________________________________________________ -->

<!-- ____________________________________________________________ -->
<!--                                                                  -->
<!-- XMI is the top-level XML element for XMI transfer text.   -->
<!-- ____________________________________________________________ -->

<!ELEMENT XMI (XMI.header, XMI.content, XMI.extensions*) >
<!ATTLIST XMI
              xmi-version CDATA #FIXED "1.0"
              timestamp   CDATA #IMPLIED
              verified (true | false) #IMPLIED
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- XMI.header contains metadata about the transfer and data about -->
<!-- the metamodels which define the content of the transfer.        -->
<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.header (XMI.documentation?, XMI.metamodel+) >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- documentation contains data about the information being transferred.  -->
<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.documentation (#PCDATA |
                             XMI.owner | XMI.contact |
                             XMI.longDescription |
                             XMI.shortDescription | XMI.exporter |
                             XMI.exporterVersion | XMI.notice)* >

<!ELEMENT XMI.owner ANY >
<!ELEMENT XMI.contact ANY >
<!ELEMENT XMI.longDescription ANY >
<!ELEMENT XMI.shortDescription ANY >
<!ELEMENT XMI.exporter ANY >
<!ELEMENT XMI.exporterVersion ANY >
<!ELEMENT XMI.exporterID ANY >
<!ELEMENT XMI.notice ANY >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- metamodel contains data about the metamodel(s) to which the content   -->
<!-- conforms.                                                            -->
<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.metamodel ANY>
<!ATTLIST XMI.metamodel
                name     CDATA #REQUIRED
                version CDATA #REQUIRED
```

```
                href     CDATA #IMPLIED
>

<!-- ____________________________________________________ -->
<!--                                                      -->
<!-- content is the actual data being transferred.    -->
<!-- ____________________________________________________ -->

<!ELEMENT XMI.content ANY >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- extensions contains information related to the content that is not   -->
<!-- defined in the metamodel(s) in the header.                           -->
<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.extensions ANY >

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- This section contains the XMI.reference element declaration.  It    -->
<!-- can be either an internal reference, in which case the target        -->
<!-- attribute contains the ID of the XML element being referred to,      -->
<!-- or it can be a reference to an XML element in another document,      -->
<!-- in which case the href attribute holds the URI of the document       -->
<!-- with an XPointer at the end.  When the XLink working draft becomes   -->
<!-- a W3C recommendation, this element will be redefined to be a         -->
<!-- simple XLink.                                                        -->

<!-- The expectedType attribute may contain the expected type of the     -->
<!-- XML element to be referred to.  Although XML processors will not    -->
<!-- enforce that the XML element referred to is the expected type,       -->
<!-- tools may report a warning if the expectedType does not match        -->
<!-- what is being referred to.                                           -->

<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.reference ANY >
<!ATTLIST XMI.reference
               target IDREF #IMPLIED
               href   CDATA #IMPLIED
               expectedType CDATA #IMPLIED
               content-title CDATA #IMPLIED
>

<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- This section contains the data types which are defined in the MOF. -->
<!-- ____________________________________________________________________ -->

<!ELEMENT XMI.field ANY >
<!ELEMENT XMI.struct (field)+ >

<!ELEMENT XMI.seqItem ANY >
<!ELEMENT XMI.sequence (seqItem)* >
```

```
<!ELEMENT XMI.arrayLen ANY >
<!ELEMENT XMI.arrayItem ANY >
<!ELEMENT XMI.array (XMI.arrayLen, XMI.arrayItem*) >

<!ELEMENT XMI.enum (#PCDATA) >

<!ELEMENT XMI.discrim ANY >
<!ELEMENT XMI.union (XMI.discrim, XMI.field*) >

<!ELEMENT XMI.any ANY >
<!ATTLIST XMI.any
              type CDATA #IMPLIED
>


<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--      This section defines common XMI entities and elements.          -->
<!-- ____________________________________________________________________ -->

<!ENTITY % XMI.elementAttributes 'XMI.id ID #REQUIRED
                                  XMI.remote (true | false) "false" ' >

<!ELEMENT XMI.extension ANY >

<!ELEMENT XMI.remoteContent (XMI.reference) >


<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!--                           Types                                      -->
<!-- This section contains the declarations of the entities and           -->
<!-- elements corresponding to the MOF Data Types.  The MOF standard      -->
<!-- enumerations are defined here, because they are used to define       -->
<!-- the rest of MOF, and XML entities must be declared before being      -->
<!-- used in a DTD.                                                       -->
<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->

<!ENTITY % AggregationType
          ' XMI.value (none | shared | composite) #REQUIRED ' >

<!ENTITY % Boolean
          ' XMI.value (true|false) #REQUIRED ' >

<!ENTITY % EvaluationType
          ' XMI.value (immediate | deferrable | deferred) #REQUIRED ' >

<!ENTITY % DirectionType
          ' XMI.value (in | out | inout | return) #REQUIRED ' >

<!ENTITY % TristateType
          ' XMI.value (yes | no | dont_care) #REQUIRED ' >
```

```
<!ENTITY % VisibilityType
          ' XMI.value (public | protected | private) #REQUIRED ' >

<!ENTITY % modelElements '(ModelElement |
                       Namespace |
                       GeneralizableElement |
                       TypedElement |
                       Classifier |
                       Class |
                       DataType |
                       Feature |
                       StructuralFeature |
                       Attribute |
                       Reference |
                       BehavioralFeature |
                       Operation |
                       Exception |
                       Association |
                       AssociationEnd |
                       Package |
                       Import |
                       Parameter |
                       Constraint |
                       Constant |
                       Tag |
                       TypeAlias )' >


<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--      This section defines associationEnds from Section 3.8.          -->
<!-- ____________________________________________________________________ -->


<!-- *********     3.8.1 Namespace-Contains-ModelElement p. 3-78 ********* -->

<!ELEMENT container (XMI.reference) >

<!ELEMENT containedElement ((%modelElements;)*) >


<!-- *********     3.8.2 GeneralizableElement-Generalizes-GeneralizableElement p.
3-80 ********* -->

<!ELEMENT superType (XMI.reference*)>

<!ELEMENT subType (XMI.reference*)>


<!-- *********     3.8.3 Reference-RefersTo-AssociationEnd p. 3-83 ********* -->

<!ELEMENT referencedEnd (XMI.reference)>

<!ELEMENT referent (XMI.reference)>
```

```
<!-- *********     3.8.4 Reference-Exposes-AssociationEnd p. 3-84 ********* -->

<!ELEMENT referrer (XMI.reference)>

<!ELEMENT exposedEnd (XMI.reference)>


<!-- *********     3.8.5 TypedElement-isOfType-Classifier p. 3-86 ********* -->

<!ELEMENT type (#PCDATA |
             XMI.reference |
                   XMI.struct |
                   XMI.sequence |
                   XMI.enum |
                   XMI.array |
                   XMI.union |
                   XMI.any)* >

<!ELEMENT typedElements (XMI.reference*) >


<!-- *********     3.8.6 Operations-CanRaise-Exception p. 3-88 ********* -->

<!ELEMENT operation (XMI.reference*) >

<!ELEMENT exception (XMI.reference*) >


<!-- *********     3.8.7 Import-Aliases-Namespace p. 3-90 ********* -->

<!ELEMENT importer (XMI.reference*) >

<!ELEMENT imported (XMI.reference*) >


<!-- *********     3.8.8 Constraint-Constrains-ModelElement p. 3-91 ********* -->

<!ELEMENT constraint (XMI.reference*) >

<!ELEMENT constrainedElement (XMI.reference*) >


<!-- *********     3.8.9 ModelElement-DependsOn-ModelElement p. 3-93 ********* -->

<!ELEMENT dependent (XMI.reference) >

<!ELEMENT provider (XMI.reference) >


<!-- *********     3.8.10 Tag-AttachesTo-ModelElement p. 3-95 ********* -->

<!ELEMENT modelElements (XMI.reference*) >
```

```
<!ELEMENT tag (XMI.reference*) >


<!-- ____________________________________________________________________ -->
<!-- ____________________________________________________________________ -->
<!--                                                                      -->
<!-- This section contains the declarations of the entities and           -->
<!-- elements corresponding to the MOF in Section 3.7.                    -->
<!-- ____________________________________________________________________ -->

<!-- *********     3.7.1 ModelElement, p. 3-13 ********* -->

<!ELEMENT name (#PCDATA) >

<!ELEMENT annotation (#PCDATA) >

<!ELEMENT qualifiedName (#PCDATA) >

<!ENTITY % ModelElementProperties 'name,
                              annotation,
                              qualifiedName+' >

<!ENTITY % ModelElementAssociations 'dependent*,
                                 provider*,
                                 container?,
                                 constraint*,
                                 tag*' >

<!ENTITY % ModelElementCompositions '' >


<!ELEMENT ModelElement (XMI.remoteContent |
                    (%ModelElementProperties;,
                    %ModelElementAssociations;,
                    %ModelElementCompositions;
                    XMI.extension*) ) >
<!ATTLIST ModelElement %XMI.elementAttributes; >



<!-- *********     3.7.2 Namespace, p. 3-21 ********* -->

<!ENTITY % NamespaceProperties '%ModelElementProperties;' >

<!ENTITY % NamespaceAssociations '%ModelElementAssociations;,
                              importer* ' >

<!ENTITY % NamespaceCompositions '%ModelElementCompositions;
                              containedElement*' >


<!ELEMENT Namespace (XMI.remoteContent |
                    (%NamespaceProperties;,
                    %NamespaceAssociations;,
```

```
                    %NamespaceCompositions;,
                    XMI.extension*) ) >
<!ATTLIST Namespace %XMI.elementAttributes; >


<!-- *********     3.7.3 GeneralizableElement, p. 3-24 ********* -->

<!ENTITY % GeneralizableElementProperties '%NamespaceProperties;,
                                    visibility,
                                    isAbstract,
                                    isRoot,
                                    isLeaf,
                                    allSuperTypes*' >

<!ENTITY % GeneralizableElementAssociations '%NamespaceAssociations;,
                                       superType*,
                                       subType*' >

<!ENTITY % GeneralizableElementCompositions '%NamespaceCompositions;' >


<!ELEMENT GeneralizableElement (XMI.remoteContent |
                    (%GeneralizableElementProperties;,
                    %GeneralizableElementAssociations;,
                    %GeneralizableElementCompositions;,
                    XMI.extension*) ) >
<!ATTLIST GeneralizableElement %XMI.elementAttributes; >

<!ELEMENT visibility EMPTY>
<!ATTLIST visibility %VisibilityType;>

<!ELEMENT isAbstract EMPTY>
<!ATTLIST isAbstract %Boolean;>

<!ELEMENT isRoot EMPTY>
<!ATTLIST isRoot %TristateType;>

<!ELEMENT isLeaf EMPTY>
<!ATTLIST isLeaf %TristateType;>

<!ELEMENT allsuperTypes (XMI.reference*)>

<!-- *********     3.7.4 TypedElement, p. 3-30 ********* -->

<!ENTITY % TypedElementProperties '%ModelElementProperties;' >

<!ENTITY % TypedElementAssociations '%ModelElementAssociations;,
                                  type' >

<!ENTITY % TypedElementCompositions '%ModelElementCompositions;' >


<!ELEMENT TypedElement (XMI.remoteContent |
                    (%TypedElementProperties;,
                    %TypedElementAssociations;,
```

```
                    %TypedElementCompositions;
                    XMI.extension*) ) >
<!ATTLIST TypedElement %XMI.elementAttributes; >


<!-- *********     3.7.5 Classifier, p. 3-32 ********* -->

<!ENTITY % ClassifierProperties '%GeneralizableElementProperties;' >

<!ENTITY % ClassifierAssociations '%GeneralizableElementAssociations;' >

<!ENTITY % ClassifierCompositions '%GeneralizableElementCompositions;' >


<!ELEMENT Classifier (XMI.remoteContent |
                    (%ClassifierProperties;,
                    %ClassifierAssociations;,
                    %ClassifierCompositions;,
                    XMI.extension*) ) >
<!ATTLIST Classifier %XMI.elementAttributes; >


<!-- *********     3.7.6 Class, p. 3-33 ********* -->

<!ENTITY % ClassProperties '%ClassifierProperties;,
                            isSingleton' >

<!ENTITY % ClassAssociations '%ClassifierAssociations;' >

<!ENTITY % ClassCompositions '%ClassifierCompositions;' >


<!ELEMENT Class (XMI.remoteContent |
                    (%ClassProperties;,
                    %ClassAssociations;,
                    %ClassCompositions;,
                    XMI.extension*) ) >
<!ATTLIST Class %XMI.elementAttributes; >

<!ELEMENT isSingleton EMPTY>
<!ATTLIST isSingleton %Boolean;>


<!-- *********     3.7.7 DataType, p. 3-36 ********* -->

<!ENTITY % DataTypeProperties '%ClassifierProperties;,
                            typeCode' >

<!ENTITY % DataTypeAssociations '%ClassifierAssociations;' >

<!ENTITY % DataTypeCompositions '%ClassifierCompositions;' >


<!ELEMENT DataType (XMI.remoteContent |
                    (%DataTypeProperties;,
```

```
                    %DataTypeAssociations;,
                    %DataTypeCompositions;,
                    XMI.extension*) ) >
<!ATTLIST DataType %XMI.elementAttributes; >

<!ELEMENT typeCode (#PCDATA)>

<!-- *********     3.7.8 Feature, p. 3-39 ********* -->

<!ENTITY % FeatureProperties '%ModelElementProperties;,
                          visibility' >

<!ENTITY % FeatureAssociations '%ModelElementAssociations;' >

<!ENTITY % FeatureCompositions '%ModelElementCompositions;' >


<!ELEMENT Feature (XMI.remoteContent |
                    (%FeatureProperties;,
                    %FeatureAssociations;,
                    %FeatureCompositions;
                    XMI.extension*) ) >
<!ATTLIST Feature %XMI.elementAttributes; >


<!-- *********     3.7.9 StructuralFeature, p. 3-41 ********* -->
<!-- Multiple inheritance from both Feature and TypedElement.  -->
<!-- Only inherit ModelElement once.  -->

<!ENTITY % StructuralFeatureProperties '%FeatureProperties;,
                                multiplicity,
                                isChangable' >

<!ENTITY % StructuralFeatureAssociations '%TypedElementAssociations;' >

<!ENTITY % StructuralFeatureCompositions '%FeatureCompositions;' >

<!ELEMENT StructuralFeature (XMI.remoteContent |
                    (%StructuralFeatureProperties;,
                    %StructuralFeatureAssociations;,
                    %StructuralFeatureCompositions;
                    XMI.extension*) ) >
<!ATTLIST StructuralFeature %XMI.elementAttributes; >

<!ELEMENT multiplicity EMPTY>
<!ATTLIST multiplicity lower CDATA #REQUIRED upper CDATA #REQUIRED isOrdered
(true|false) #IMPLIED isUnique (true|false) #IMPLIED >

<!ELEMENT isChangable EMPTY>
<!ATTLIST isChangable %Boolean; >


<!-- *********     3.7.10 Attribute, p. 3-43 ********* -->

<!ENTITY % AttributeProperties '%StructuralFeatureProperties;,
```

```
                          isDerived ' >

<!ENTITY % AttributeAssociations '%StructuralFeatureAssociations;' >

<!ENTITY % AttributeCompositions '%StructuralFeatureCompositions;' >


<!ELEMENT Attribute (XMI.remoteContent |
                    (%AttributeProperties;,
                    %AttributeAssociations;,
                    %AttributeCompositions;
                    XMI.extension*) ) >
<!ATTLIST Attribute %XMI.elementAttributes; >

<!ELEMENT isDerived EMPTY>
<!ATTLIST isDerived %Boolean; >


<!-- *********     3.7.11 Reference, p. 3-44 ********* -->

<!ENTITY % ReferenceProperties '%StructuralFeatureProperties;' >

<!ENTITY % ReferenceAssociations '%StructuralFeatureAssociations;,
                              referencedEnd,
                              exposedEnd ' >

<!ENTITY % ReferenceCompositions '%StructuralFeatureCompositions;' >


<!ELEMENT Reference (XMI.remoteContent |
                    (%ReferenceProperties;,
                    %ReferenceAssociations;,
                    %ReferenceCompositions;
                    XMI.extension*) ) >
<!ATTLIST Reference %XMI.elementAttributes; >


<!-- *********     3.7.12 BehavioralFeature, p. 3-47 ********* -->
<!-- Multiple inheritance from both Feature and Namespace.  -->
<!-- Only inherit ModelElement once.  -->

<!ENTITY % BehavioralFeatureProperties '%NamespaceProperties;' >

<!ENTITY % BehavioralFeatureAssociations '%NamespaceAssociations;' >

<!ENTITY % BehavioralFeatureCompositions '%NamespaceCompositions;' >


<!ELEMENT BehavioralFeature (XMI.remoteContent |
                    (%BehavioralFeatureProperties;,
                    %BehavioralFeatureAssociations;,
                    %BehavioralFeatureCompositions;,
                    XMI.extension*) ) >
<!ATTLIST BehavioralFeature %XMI.elementAttributes; >
```

```
<!-- *********     3.7.13 Operation, p. 3-44 ********* -->

<!ENTITY % OperationProperties '%BehavioralFeatureProperties;,
                          isQuery ' >

<!ENTITY % OperationAssociations '%BehavioralFeatureAssociations;,
                             exceptions* ' >

<!ENTITY % OperationCompositions '%BehavioralFeatureCompositions;' >


<!ELEMENT Operation (XMI.remoteContent |
                    (%OperationProperties;,
                    %OperationAssociations;,
                    %OperationCompositions;,
                    XMI.extension*) ) >
<!ATTLIST Operation %XMI.elementAttributes; >

<!ELEMENT isQuery EMPTY>
<!ATTLIST isQuery %Boolean; >


<!-- *********     3.7.14 Exception, p. 3-51 ********* -->

<!ENTITY % ExceptionProperties '%BehavioralFeatureProperties; ' >

<!ENTITY % ExceptionAssociations '%BehavioralFeatureAssociations;,
                             operation* ' >

<!ENTITY % ExceptionCompositions '%BehavioralFeatureCompositions;' >


<!ELEMENT Exception (XMI.remoteContent |
                    (%ExceptionProperties;,
                    %ExceptionAssociations;,
                    %ExceptionCompositions;,
                    XMI.extension*) ) >
<!ATTLIST Exception %XMI.elementAttributes; >


<!-- *********     3.7.15 Association, p. 3-53 ********* -->

<!ENTITY % AssociationProperties '%ClassifierProperties;,
                          isDerived ' >

<!ENTITY % AssociationAssociations '%ClassifierAssociations;' >

<!ENTITY % AssociationCompositions '%ClassifierCompositions;,
                             source,
                             target ' >


<!ELEMENT Association (XMI.remoteContent |
                    (%AssociationProperties;,
```

```
                    %AssociationAssociations;,
                    %AssociationCompositions;,
                    XMI.extension*) ) >
<!ATTLIST Association %XMI.elementAttributes; >

<!ELEMENT source (AssociationEnd)>

<!ELEMENT target (AssociationEnd)>


<!-- *********     3.7.16 AssociationEnd, p. 3-56 ********* -->
<!-- Multiple inheritance from both Feature and TypedElement.  -->
<!-- Only inherit ModelElement once.  -->

<!ENTITY % AssociationEndProperties '%TypedElementProperties;,
                                    multiplicity,
                                    aggregation,
                                    isNaviagble,
                                    isChangable,
                                    otherEnd ' >

<!ENTITY % AssociationEndAssociations '%TypedElementAssociations;,
                                    referent?,
                                    referrer?' >

<!ENTITY % AssociationEndCompositions '%TypedElementCompositions;' >


<!ELEMENT AssociationEnd (XMI.remoteContent |
                    (%AssociationEndProperties;,
                    %AssociationEndAssociations;,
                    %AssociationEndCompositions;
                    XMI.extension*) ) >
<!ATTLIST AssociationEnd %XMI.elementAttributes; >

<!ELEMENT aggregation EMPTY>
<!ATTLIST aggregation %AggregationType; >

<!ELEMENT isNaviagble EMPTY>
<!ATTLIST isNaviagble %Boolean; >

<!ELEMENT otherEnd (XMI.reference)>


<!-- *********     3.7.17 Package, p. 3-32 ********* -->

<!ENTITY % PackageProperties '%GeneralizableElementProperties;' >

<!ENTITY % PackageAssociations '%GeneralizableElementAssociations;' >

<!ENTITY % PackageCompositions '%GeneralizableElementCompositions;' >


<!ELEMENT Package (XMI.remoteContent |
                    (%PackageProperties;,
```

```
                    %PackageAssociations;,
                    %PackageCompositions;,
                    XMI.extension*) ) >
<!ATTLIST Package %XMI.elementAttributes; >


<!-- *********     3.7.18 Import, p. 3-64 ********* -->

<!ENTITY % ImportProperties '%ModelElementProperties;,
                            visibility' >

<!ENTITY % ImportAssociations '%ModelElementAssociations;,
                            imported ' >

<!ENTITY % ImportCompositions '%ModelElementCompositions;' >


<!ELEMENT Import (XMI.remoteContent |
                    (%ImportProperties;,
                    %ImportAssociations;,
                    %ImportCompositions;
                    XMI.extension*) ) >
<!ATTLIST Import %XMI.elementAttributes; >


<!-- *********     3.7.19 Parameter, p. 3-67 ********* -->

<!ENTITY % ParameterProperties '%TypedElementProperties;,
                            direction,
                            multiplicity ' >

<!ENTITY % ParameterAssociations '%TypedElementAssociations; ' >

<!ENTITY % ParameterCompositions '%TypedElementCompositions;' >


<!ELEMENT Parameter (XMI.remoteContent |
                    (%ParameterProperties;,
                    %ParameterAssociations;,
                    %ParameterCompositions;
                    XMI.extension*) ) >
<!ATTLIST Parameter %XMI.elementAttributes; >

<!ELEMENT direction EMPTY>
<!ATTLIST direction %DirectionType; >


<!-- *********     3.7.20 Constraint, p. 3-69 ********* -->

<!ENTITY % ConstraintProperties '%ModelElementProperties;,
                            expression,
                            language,
                            evaluationPolicy ' >

<!ENTITY % ConstraintAssociations '%ModelElementAssociations;,
```

```
                              constrainedElements+ ' >

<!ENTITY % ConstraintCompositions '%ModelElementCompositions;' >


<!ELEMENT Constraint (XMI.remoteContent |
                    (%ConstraintProperties;,
                    %ConstraintAssociations;,
                    %ConstraintCompositions;
                    XMI.extension*) ) >
<!ATTLIST Constraint %XMI.elementAttributes; >

<!ELEMENT expression (XMI.any)>

<!ELEMENT language (#PCDATA)>

<!ELEMENT evaluationPolicy EMPTY>
<!ATTLIST evaluationPolicy %EvaluationType; >


<!-- *********     3.7.21 Constant, p. 3-73 ********* -->

<!ENTITY % ConstantProperties '%TypedElementProperties;,
                            value ' >

<!ENTITY % ConstantAssociations '%TypedElementAssociations; ' >

<!ENTITY % ConstantCompositions '%TypedElementCompositions;' >


<!ELEMENT Constant (XMI.remoteContent |
                    (%ConstantProperties;,
                    %ConstantAssociations;,
                    %ConstantCompositions;
                    XMI.extension*) ) >
<!ATTLIST Constant %XMI.elementAttributes; >

<!ELEMENT value ANY>


<!-- *********     3.7.22 Tag, p. 3-74 ********* -->

<!ENTITY % TagProperties '%ModelElementProperties;,
                            tagId,
                            values* ' >

<!ENTITY % TagAssociations '%ModelElementAssociations;,
                            modelElements+ ' >

<!ENTITY % TagCompositions '%ModelElementCompositions;' >


<!ELEMENT Tag (XMI.remoteContent |
                    (%TagProperties;,
                    %TagAssociations;,
```

```
                    %TagCompositions;
                    XMI.extension*) ) >
<!ATTLIST Tag %XMI.elementAttributes; >

<!ELEMENT tagId (#PCDATA)>

<!ELEMENT values ANY>


<!-- *********     3.7.23 TypeAlias p. 3-77 ********* -->

<!ENTITY % TypeAliasProperties '%TypedElementProperties;,
                          multiplicity ' >

<!ENTITY % TypeAliasAssociations '%TypedElementAssociations; ' >

<!ENTITY % TypeAliasCompositions '%TypedElementCompositions;' >


<!ELEMENT TypeAlias (XMI.remoteContent |
                    (%TypeAliasProperties;,
                    %TypeAliasAssociations;,
                    %TypeAliasCompositions;
                    XMI.extension*) ) >
<!ATTLIST TypeAlias %XMI.elementAttributes; >

<!--          End of Document      -->
```

B.3 Example

This is an example MOF document which consists of a package containing a class that contains an attribute.

```
<?xml version="1.0"?>
<!DOCTYPE XMI SYSTEM "mof.dtd">

<XMI xmi-version="1.0">
   <XMI.header>
      <XMI.metamodel name="uml" version="1.1" />
   </XMI.header>
   <XMI.content>
      <Package XMI.id="i00000001">
         <name>package1</name>
         <annotation/>
         <qualifiedName/>
         <visibility XMI.value="public" />
         <isAbstract XMI.value="false"/>
         <isRoot XMI.value="yes"/>
         <isLeaf XMI.value="yes"/>
         <containedElement>
            <Class XMI.id="i00000002">
               <name>class1</name>
               <annotation/>
               <qualifiedName/>
```

```
                    <visibility XMI.value="public" />
                    <isAbstract XMI.value="false"/>
                    <isRoot XMI.value="yes"/>
                    <isLeaf XMI.value="yes"/>
                    <isSingleton XMI.value="false"/>
                    <containedElement>
                       <Attribute XMI.id="i00000003">
                          <name>attribute1</name>
                          <annotation/>
                          <qualifiedName/>
                          <visibility XMI.value="public" />
                          <multiplicity lower="1" upper="1"/>
                          <isChangable XMI.value="true" />
                          <isDerived XMI.value="false" />
                          <type>integer</type>
                       </Attribute>
                    </containedElement>
                 </Class>
              </containedElement>
           </Package>
      </XMI.content>
</XMI>
```

Example Model Encodings C

C.1 Introduction

This Appendix provides two extremely simple UML models and their resultant XML encoding.

C.2 Model1 Model

This is an example of a model containing a class which contains an attribute. It is encoded in XMI using the UML DTD.

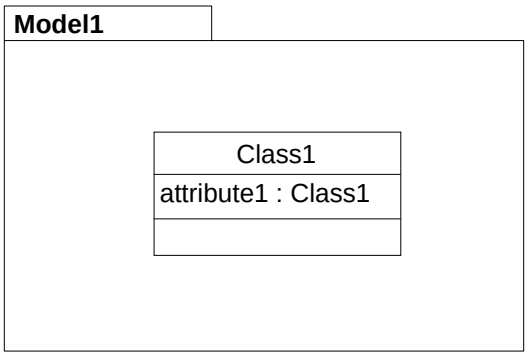

Figure C-1 Simple UML Model Model1

```
<?xml version="1.0"?>
<!DOCTYPE XMI SYSTEM "uml.dtd">
```

```
<XMI xmi-version="1.0">
    <XMI.header>
      <XMI.metamodel name="uml" version="1.1" />
    </XMI.header>
    <XMI.content>
      <Model XMI.id="i00000001">
        <name>model1</name>
        <visibility XMI.value="public"/>
        <isAbstract XMI.value="false"/>
        <isLeaf XMI.value="true"/>
        <isRoot XMI.value="true"/>
        <ownedElement>
          <Class XMI.id="i00000002">
            <name>class1</name>
            <visibility XMI.value="public"/>
            <isAbstract XMI.value="false"/>
            <isLeaf XMI.value="true"/>
            <isRoot XMI.value="true"/>
            <isActive XMI.value="true"/>
            <feature>
              <Attribute XMI.id="i00000003">
                <name>attribute1</name>
                <visibility XMI.value="public"/>
                <ownerScope XMI.value="classifier"/>
                <changeable XMI.value="none"/>
                <multiplicity>1</multiplicity>
                <initialValue>0</initialValue>
                <owner>
                  <XMI.reference href="|i00000002" expected-Type="class1"/>
                </owner>
                <type>integer</type>
              </Attribute>
            </feature>
          </Class>
        </ownedElement>
      </Model>
    </XMI.content>
</XMI>
```

C.3 Business Model

The following Model was developed as the source for a simple XML document produced according to XMI.

The XML resulting from this model is shown below. Note that some discrepancies occur between this prototyped implementation and the current XML production rules.

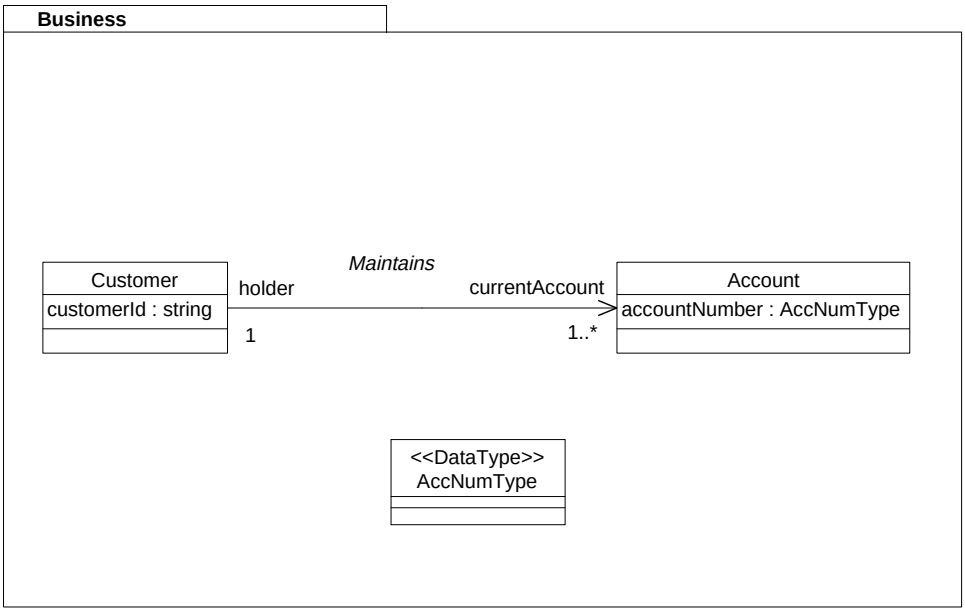

Figure C-2 Simple UML Model Business

```
<?XML version = "1.0"?>
<!DOCTYPE System SYSTEM "file:unisys-uml.dtd">
<System elementVisibility = 'public' InventedStart= '{' InventedEnd = '}' id = 'G.6'>
  <!-- name: {Rose 98 System} -->
 <Model elementVisibility = 'public' id = 'G.1'>
  <name>BusinessModel</name>
  <Enumeration elementVisibility = 'public' id = 'G.7'>
   <name>boolean</name>
   <EnumerationLiteral>
    <name>false</name>
   </EnumerationLiteral>
   <EnumerationLiteral>
    <name>true</name>
   </EnumerationLiteral>
  </Enumeration>
  <DataType elementVisibility = 'public' id = 'G.8'>
   <name>byte</name>
  </DataType>
  <DataType elementVisibility = 'public' id = 'G.9'>
   <name>currency</name>
  </DataType>
  <DataType elementVisibility = 'public' id = 'G.10'>
   <name>date</name>
  </DataType>
  <DataType elementVisibility = 'public' id = 'G.11'>
   <name>double</name>
  </DataType>
  <DataType elementVisibility = 'public' id = 'G.12'>
```

```
  <name>integer</name>
</DataType>
<DataType elementVisibility = 'public' id = 'G.13'>
  <name>long</name>
</DataType>
<DataType elementVisibility = 'public' id = 'G.14'>
  <name>object</name>
</DataType>
<DataType elementVisibility = 'public' id = 'G.15'>
  <name>single</name>
</DataType>
<DataType elementVisibility = 'public' id = 'G.16'>
  <name>string</name>
</DataType>
<DataType elementVisibility = 'public' id = 'G.17'>
  <name>variant</name>
</DataType>
<Package id = 'S.10006'>
  <name>Business</name>
</Package>
<Class id = 'S.10001'>
  <name>Customer</name>
  <isAbstract Boolean = 'false'/>
  <isLeaf Boolean = 'true'/>
  <isRoot Boolean = 'true'/>
  <associationEndRef reference = 'G.5'/> <!-- holder{359BFCB10337} -->
  <feature>
    <Attribute id = 'S.10002' elementVisibility = 'private'>
      <name>customerId</name>
      <ownerScope ScopeKind = 'instance'/>
      <type reference = 'G.16'/> <!-- string -->
    </Attribute>
  </feature>
</Class>
<Class id = 'S.10003'>
  <name>Account</name>
  <isAbstract Boolean = 'false'/>
  <isLeaf Boolean = 'true'/>
  <isRoot Boolean = 'true'/>
  <associationEndRef reference = 'G.4'/> <!-- currentAccount{359BFCB1032D} -->
  <feature>
    <Attribute id = 'S.10004' elementVisibility = 'private'>
      <name>accountNumber</name>
      <ownerScope ScopeKind = 'instance'/>
      <type reference = 'S.10005'/> <!-- AccNumType -->
    </Attribute>
  </feature>
</Class>
<Class id = 'S.10005'>
  <name>AccNumType</name>
  <stereotypeRef reference = 'G.2'/> <!-- DataType -->
```

```
      <isAbstract Boolean = 'false'/>
      <isLeaf Boolean = 'true'/>
      <isRoot Boolean = 'true'/>
    </Class>
    <Association id = 'G.3'>
      <name>Maintains{359BFCAF0302}</name>
      <connection>
        <AssociationEnd id = 'G.4'>
          <name>currentAccount{359BFCB1032D}</name>
          <multiplicity>1..*</multiplicity>
          <endpointType reference = 'S.10003'/> <!-- Logical View::Account -->
          <isNavigable Boolean = 'true'/>
          <aggregation AggregationKind = 'none'/>
        </AssociationEnd>
        <AssociationEnd id = 'G.5'>
          <name>holder{359BFCB10337}</name>
          <multiplicity>1..1</multiplicity>
          <endpointType reference = 'S.10001'/> <!-- Logical View::Customer -->
          <isNavigable Boolean = 'false'/>
          <aggregation AggregationKind = 'none'/>
        </AssociationEnd>
      </connection>
    </Association>
    <Stereotype id = 'G.2'>
      <name>DataType</name>
      <baseClass>Class</baseClass>
      <extendedElement reference = 'S.10005'/>
    </Stereotype>
  </Model>
</System>
```