

XML Metadata Interchange (XMI)

Appendices

Version 1.1

Joint Submission

<i>Unisys Corporation</i>	<i>Fujitsu</i>
<i>International Business Machines Corporation</i>	<i>Softeam</i>
<i>Cooperative Research Centre for Distributed Systems Technology (DSTC)</i>	<i>Recerca Informatica</i>
<i>Oracle Corporation</i>	<i>Daimler-Benz</i>
<i>Platinum Technology, Inc.</i>	

Supported by:

<i>Cayenne Software</i>	<i>Ardent</i>
<i>Genesis Development</i>	<i>Aviatis</i>
<i>Inline Software</i>	<i>ICONIX</i>
<i>Rational Software Corporation</i>	<i>Integrated Systems</i>
<i>Select Software</i>	<i>Verilog</i>
<i>Sprint Communications Company</i>	<i>Telefonica I+D</i>
<i>Sybase, Inc.</i>	<i>Universitat Politecnica de Catalunya</i>
<i>Xerox</i>	<i>NCR</i>
<i>EDS</i>	<i>Nihon Unisys</i>
<i>Boeing</i>	<i>NTT</i>

OMG Document ad/99-10-03

October 25, 1999

Copyright 1998, 1999 Unisys Corporation
Copyright 1998, 1999 IBM Corporation
Copyright 1998, 1999 Cooperative Research Centre for Distributed Systems Technology (DSTC)
Copyright 1998, 1999 Oracle Corporation
Copyright 1998, 1999 Platinum Technology, Inc.
Copyright 1998, 1999 Fujitsu
Copyright 1998, 1999 Softeam
Copyright 1998, 1999 Recerca Informatica
Copyright 1998, 1999 Daimler-Benz

The companies listed above hereby grant a royalty-free license to the Object Management Group, Inc. (OMG) for worldwide distribution of this document or any derivative works thereof, so long as the OMG reproduces the copyright notices and the below paragraphs on all distributed copies.

The material in this document is submitted to the OMG for evaluation. Submission of this document does not represent a commitment to implement any portion of this specification in the products of the submitters.

WHILE THE INFORMATION IN THIS PUBLICATION IS BELIEVED TO BE ACCURATE, THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. The companies listed above shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. The information contained in this document is subject to change without notice.

This document contains information which is protected by copyright. All Rights Reserved. Except as otherwise provided herein, no part of this work may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without the permission of one of the copyright owners. All copies of this document must include the copyright and other information contained on this page.

The copyright owners grant member companies of the OMG permission to make a limited number of copies of this document (up to fifty copies) for their internal use as part of the OMG evaluation process.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Right in Technical Data and Computer Software Clause at DFARS 252.227.7013.

CORBA, OMG, and Object Request Broker are trademarks of Object Management Group.

Table of Contents

Appendix A	UML 1.1 DTD	A-5
A.1	Introduction	A-5
A.2	UML DTD	A-5
Appendix B	MOF 1.1 DTD	B-123
B.1	Introduction	B-123
B.2	MOF DTD	B-123
Appendix C	Example Model Encodings.....	C-143
C.1	Introduction	C-143
C.2	Example 1 - Department.....	C-143
C.3	Example 2 - Campus and Schedule	C-145
C.4	Example 3 - Instances of Schedule	C-148
C.5	Example 4 - “Complete” datatype metamodel.	C-150

A.1 Introduction

This appendix contains a normative DTD that represents the UML 1.1 metamodel. It was generated by a program that closely follows rule set 1 in Chapter 7, “XML DTD Production”. The program created the DTD from a Rational Rose model representing the UML 1.1 metamodel as defined in the document “UML Semantics version 1.1, 1 September 1997”. The metamodel in that document was changed by adding role names to associations that did not have them.

As noted in Section 4.4.12 of the XMI submission, the XMI submitters are not attempting to create official metamodels for UML (or for the MOF). We propose that the groups that propose and maintain the standards for metamodels create official versions of the metamodels and DTDs corresponding to them.

In the meantime, there is a need to exchange MOF and UML metadata now, so this appendix presents a normative DTD for UML 1.1; it can be used to develop software.

On minor change from rule set 1 is that all composed roles are included at the beginning of the UML declarations in the DTD, whereas rule set 1 would place them in the packages that they belong to. That difference does not change the semantics of the DTD.

The source of the UML 1.1 metamodel may have outstanding issues that might affect the DTD included in this appendix.

A.2 UML DTD

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

<!-- Generated by: XMI Framework 1.0 -->
<!-- Date: Fri Oct 22 14:31:09 PDT 1999 -->
```

```

<!-- _____ -->
<!-- _____ -->
<!-- This section of the DTD contains XML declarations required by -->
<!-- XMI. -->
<!-- _____ -->

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

<!ELEMENT XMI (XMI.header?, XMI.content?, XMI.difference*,
               XMI.extensions*) >
<!ATTLIST XMI
           xmi.version CDATA #FIXED "1.1"
           timestamp CDATA #IMPLIED
           verified (true | false) #IMPLIED
>

<!-- _____ -->
<!-- _____ -->
<!-- XMI.header contains documentation and identifies the model, -->
<!-- metamodel, and metamodel -->
<!-- _____ -->

<!ELEMENT XMI.header (XMI.documentation?, XMI.model*, XMI.metamodel*,
                     XMI.metamodel*, XMI.import*) >

<!-- _____ -->
<!-- _____ -->
<!-- documentation for transfer data -->
<!-- _____ -->

<!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 >

<!-- _____ -->
<!-- _____ -->
<!-- XMI.element.att defines the attributes that each XML element -->
<!-- that corresponds to a metamodel class must have to conform to -->
<!-- the XMI specification. -->
<!-- _____ -->

```

```

<!ENTITY % XMI.element.att
    'xmi.id ID #IMPLIED xmi.label CDATA #IMPLIED xmi.uuid
    CDATA #IMPLIED ' >

<!-- _____ -->
<!--                                     -->
<!-- XMI.link.att defines the attributes that each XML element that -->
<!-- corresponds to a metamodel class must have to enable it to -->
<!-- function as a simple XLink as well as refer to model -->
<!-- constructs within the same XMI file. -->
<!-- _____ -->

<!ENTITY % XMI.link.att
    'href CDATA #IMPLIED xmi.idref IDREF #IMPLIED xml:link
    CDATA #IMPLIED xlink:inline (true | false) #IMPLIED
    xlink:actuate (show | user) #IMPLIED xlink:content-role
    CDATA #IMPLIED xlink:title CDATA #IMPLIED xlink:show
    (embed | replace | new) #IMPLIED xlink:behavior CDATA
    #IMPLIED' >

<!-- _____ -->
<!--                                     -->
<!-- XMI.model identifies the model(s) being transferred -->
<!-- _____ -->

<!ELEMENT XMI.model ANY >
<!ATTLIST XMI.model
    %XMI.link.att;
    xmi.name          CDATA #REQUIRED
    xmi.version       CDATA #IMPLIED
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.metamodel identifies the metamodel(s) for the transferred -->
<!-- data -->
<!-- _____ -->

<!ELEMENT XMI.metamodel ANY >
<!ATTLIST XMI.metamodel
    %XMI.link.att;
    xmi.name          CDATA #REQUIRED
    xmi.version       CDATA #IMPLIED
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.metametamodel identifies the metametamodel(s) for the -->
<!-- transferred data -->
<!-- _____ -->

<!ELEMENT XMI.metametamodel ANY >
<!ATTLIST XMI.metametamodel
    %XMI.link.att;

```

```

        xmi.name      CDATA #REQUIRED
        xmi.version   CDATA #IMPLIED
    >

<!-- _____ -->
<!-- _____ -->
<!-- XMI.metamodel identifies other files associated with the -->
<!-- transferred data -->
<!-- _____ -->

<!ELEMENT XMI.import ANY >
<!ATTLIST XMI.import
    %XMI.link.att;
    xmi.name      CDATA #REQUIRED
    xmi.version   CDATA #IMPLIED
>

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

<!ELEMENT XMI.content ANY >

<!-- _____ -->
<!-- _____ -->
<!-- XMI.extensions contains data to transfer that does not conform -->
<!-- to the metamodel(s) in the header -->
<!-- _____ -->

<!ELEMENT XMI.extensions ANY >
<!ATTLIST XMI.extensions
    xmi.extender CDATA #REQUIRED
>

<!-- _____ -->
<!-- _____ -->
<!-- extension contains information related to a specific model -->
<!-- construct that is not defined in the metamodel(s) in the header -->
<!-- _____ -->

<!ELEMENT XMI.extension ANY >
<!ATTLIST XMI.extension
    %XMI.element.att;
    %XMI.link.att;
    xmi.extender CDATA #REQUIRED
    xmi.extenderID CDATA #IMPLIED
>

<!-- _____ -->
<!-- _____ -->
<!-- XMI.difference holds XML elements representing differences to a -->
<!-- base model -->
<!-- _____ -->

```



```

<!ELEMENT XMI.difference (XMI.difference | XMI.delete | XMI.add |
                           XMI.replace)* >
<!ATTLIST XMI.difference
           %XMI.element.att;
           %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.delete represents a deletion from a base model -->
<!-- _____ -->

<!ELEMENT XMI.delete EMPTY >
<!ATTLIST XMI.delete
           %XMI.element.att;
           %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.add represents an addition to a base model -->
<!-- _____ -->

<!ELEMENT XMI.add ANY >
<!ATTLIST XMI.add
           %XMI.element.att;
           %XMI.link.att;
           xmi.position CDATA "-1"
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.replace represents the replacement of a model construct -->
<!-- with another model construct in a base model -->
<!-- _____ -->

<!ELEMENT XMI.replace ANY >
<!ATTLIST XMI.replace
           %XMI.element.att;
           %XMI.link.att;
           xmi.position CDATA "-1"
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.reference may be used to refer to data types not defined in -->
<!-- the metamodel -->
<!-- _____ -->

<!ELEMENT XMI.reference ANY >
<!ATTLIST XMI.reference
           %XMI.link.att;
>

<!ATTLIST XMI

```

```

xmlns:UML CDATA #IMPLIED
>

<!ELEMENT UML:Binding.argument (UML:ModelElement | UML:Comment |
                                UML:Namespace | UML:GeneralizableElement |
                                UML:Feature | UML:Parameter |
                                UML:Constraint | UML:Dependency |
                                UML:Generalization | UML:AssociationEnd |
                                UML:Request | UML:ActionSequence |
                                UML:Action | UML:Link | UML:LinkEnd |
                                UML:Instance | UML:AttributeLink |
                                UML:MessageInstance | UML:Interaction |
                                UML:Message | UML:StateMachine |
                                UML:Guard | UML:StateVertex |
                                UML:Transition | UML:Event |
                                UML:Partition | UML:Collaboration |
                                UML:Classifier | UML:Association |
                                UML:Stereotype | UML:Package |
                                UML:Signal | UML:Node | UML:Component |
                                UML:Interface | UML:Class | UML:DataType |
                                UML:Subsystem | UML:ClassifierRole |
                                UML:Actor | UML:UseCase |
                                UML:ClassifierInState |
                                UML:AssociationClass | UML:Enumeration |
                                UML:Primitive | UML:Structure |
                                UML:AssociationRole | UML:Model |
                                UML:Exception | UML:StructuralFeature |
                                UML:BehavioralFeature | UML:Attribute |
                                UML:Operation | UML:Method |
                                UML:Reception | UML:Refinement |
                                UML:Usage | UML:Trace | UML:Binding |
                                UML:AssociationEndRole |
                                UML:CreateAction | UML:CallAction |
                                UML:LocalInvocation | UML:ReturnAction |
                                UML:SendAction | UML:UninterpretedAction |
                                UML:TerminateAction | UML:DestroyAction |
                                UML:LinkObject | UML:Object |
                                UML:DataValue | UML:UseCaseInstance |
                                UML:ActivityModel | UML:PseudoState |
                                UML:State | UML:CompositeState |
                                UML:SimpleState | UML:SubmachineState |
                                UML:ActionState | UML:ObjectFlowState |
                                UML:ActivityState | UML:SignalEvent |
                                UML:CallEvent | UML:TimeEvent |
                                UML:ChangeEvent)* >

<!ELEMENT UML:ModelElement.taggedValue (UML:TaggedValue)* >

<!ELEMENT UML:Namespace.ownedElement (UML:ModelElement | UML:Comment |
                                         UML:Namespace |
                                         UML:GeneralizableElement |
                                         UML:Feature | UML:Parameter |

```

```

UML:Constraint | UML:Dependency |
UML:Generalization |
UML:AssociationEnd | UML:Request |
UML:ActionSequence | UML:Action |
UML:Link | UML:LinkEnd |
UML:Instance | UML:AttributeLink |
UML:MessageInstance |
UML:Interaction | UML:Message |
UML:StateMachine | UML:Guard |
UML:StateVertex | UML:Transition |
UML:Event | UML:Partition |
UML:Collaboration | UML:Classifier |
UML:Association | UML:Stereotype |
UML:Package | UML:Signal |
UML:Node | UML:Component |
UML:Interface | UML:Class |
UML:DataType | UML:Subsystem |
UML:ClassifierRole | UML:Actor |
UML:UseCase |
UML:ClassifierInState |
UML:AssociationClass |
UML:Enumeration | UML:Primitive |
UML:Structure |
UML:AssociationRole | UML:Model |
UML:Exception |
UML:StructuralFeature |
UML:BehavioralFeature |
UML:Attribute | UML:Operation |
UML:Method | UML:Reception |
UML:Refinement | UML:Usage |
UML:Trace | UML:Binding |
UML:AssociationEndRole |
UML:CreateAction | UML:CallAction |
UML:LocalInvocation |
UML:ReturnAction | UML:SendAction |
UML:UninterpretedAction |
UML:TerminateAction |
UML:DestroyAction | UML:LinkObject |
UML:Object | UML:DataValue |
UML:UseCaseInstance |
UML:ActivityModel |
UML:PseudoState | UML:State |
UML:CompositeState |
UML:SimpleState |
UML:SubmachineState |
UML:ActionState |
UML:ObjectFlowState |
UML:ActivityState |
UML:SignalEvent | UML:CallEvent |
UML:TimeEvent | UML:ChangeEvent)* >

<!ELEMENT UML:Classifier.feature (UML:Feature | UML:StructuralFeature |
                                UML:BehavioralFeature | UML:Attribute |
                                UML:Operation | UML:Method |

```

```

                                UML:Reception)* >

<!ELEMENT UML:BehavioralFeature.parameter (UML:Parameter)* >

<!ELEMENT UML:Dependency.subDependencies (UML:Dependency |
                                           UML:Refinement | UML:Usage |
                                           UML:Trace | UML:Binding)* >

<!ELEMENT UML:AssociationEnd.qualifier (UML:Attribute)* >

<!ELEMENT UML:Association.connection (UML:AssociationEnd |
                                       UML:AssociationEndRole)* >

<!ELEMENT UML:Stereotype.requiredTag (UML:TaggedValue)* >

<!ELEMENT UML:Enumeration.literal (UML:EnumerationLiteral)* >

<!ELEMENT UML:Signal.parameter (UML:Parameter)* >

<!ELEMENT UML:ActionSequence.action (UML:Action | UML:CreateAction |
                                       UML:CallAction |
                                       UML:LocalInvocation |
                                       UML:ReturnAction | UML:SendAction |
                                       UML:UninterpretedAction |
                                       UML:TerminateAction |
                                       UML:DestroyAction)* >

<!ELEMENT UML:Action.actualArgument (UML:Argument)* >

<!ELEMENT UML:Link.linkRole (UML:LinkEnd)* >

<!ELEMENT UML:Instance.slot (UML:AttributeLink)* >

<!ELEMENT UML:Collaboration.interaction (UML:Interaction)* >

<!ELEMENT UML:Interaction.message (UML:Message)* >

<!ELEMENT UML:StateMachine.top (UML:State | UML:CompositeState |
                                   UML:SimpleState | UML:SubmachineState |
                                   UML:ActionState | UML:ObjectFlowState |
                                   UML:ActivityState)* >

```

```

<!ELEMENT UML:StateMachine.transitions (UML:Transition)* >

<!ELEMENT UML:Transition.guard (UML:Guard)* >

<!ELEMENT UML:Transition.effect (UML:ActionSequence)* >

<!ELEMENT UML:State.internalTransition (UML:Transition)* >

<!ELEMENT UML:State.entry (UML:ActionSequence)* >

<!ELEMENT UML:State.exit (UML:ActionSequence)* >

<!ELEMENT UML:CompositeState.substate (UML:StateVertex |
                                         UML:PseudoState | UML:State |
                                         UML:CompositeState |
                                         UML:SimpleState |
                                         UML:SubmachineState |
                                         UML:ActionState |
                                         UML:ObjectFlowState |
                                         UML:ActivityState)* >

<!ELEMENT UML:ActivityModel.partition (UML:Partition)* >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Auxiliary_Elements.Refinement -->
<!-- _____ -->

<!ELEMENT UML:Refinement.mapping (UML:Mapping) >

<!ELEMENT UML:Refinement (UML:ModelElement.name |
                           UML:ModelElement.visibility |
                           UML:Dependency.description |
                           UML:Refinement.mapping | XMI:extension |
                           UML:ModelElement.binding |
                           UML:ModelElement.template |
                           UML:ModelElement.templateParameter |
                           UML:ModelElement.implementation |
                           UML:ModelElement.view |
                           UML:ModelElement.presentation |
                           UML:ModelElement.namespace |
                           UML:ModelElement.constraint |
                           UML:ModelElement.requirement |
                           UML:ModelElement.provision |
                           UML:ModelElement.stereotype |
                           UML:ModelElement.elementReference |
                           UML:ModelElement.collaboration |
                           UML:ModelElement.behavior |
                           UML:ModelElement.partition |

```

```

                                UML:Dependency.owningDependency |
                                UML:Dependency.client |
                                UML:Dependency.supplier |
                                UML:ModelElement.taggedValue |
                                UML:Dependency.subDependencies)* >
<!-- UML:Refinement
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    description CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owningDependency IDREFS #IMPLIED
    client IDREFS #IMPLIED
    supplier IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Auxiliary_Elements.Usage -->
<!-- _____ -->

<!-- UML:Usage (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:Dependency.description | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:Dependency.owningDependency |

```

```

        UML:Dependency.client | UML:Dependency.supplier |
        UML:ModelElement.taggedValue |
        UML:Dependency.subDependencies)* >
<!-- UML:Usage
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    description CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owningDependency IDREFS #IMPLIED
    client IDREFS #IMPLIED
    supplier IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Auxiliary_Elements.Trace -->
<!-- _____ -->

<!-- UML:Trace (UML:ModelElement.name |
        UML:ModelElement.visibility |
        UML:Dependency.description | XMI.extension |
        UML:ModelElement.binding |
        UML:ModelElement.template |
        UML:ModelElement.templateParameter |
        UML:ModelElement.implementation |
        UML:ModelElement.view |
        UML:ModelElement.presentation |
        UML:ModelElement.namespace |
        UML:ModelElement.constraint |
        UML:ModelElement.requirement |
        UML:ModelElement.provision |
        UML:ModelElement.stereotype |
        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition |
        UML:Dependency.owningDependency |
        UML:Dependency.client | UML:Dependency.supplier |
        UML:ModelElement.taggedValue |

```

```

                                UML:Dependency.subDependencies)* >
<!-- UML:Trace
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    description CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owningDependency IDREFS #IMPLIED
    client IDREFS #IMPLIED
    supplier IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
-->

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Auxiliary_Elements.Binding -->
<!-- _____ -->

<!-- UML:Binding (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:Dependency.description | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:Dependency.owningDependency |
    UML:Dependency.client | UML:Dependency.supplier |
    UML:ModelElement.taggedValue |
    UML:Dependency.subDependencies |
    UML:Binding.argument)* >

```



```

<!ATTLIST UML:Binding
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    description CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owningDependency IDREFS #IMPLIED
    client IDREFS #IMPLIED
    supplier IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;

>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Auxiliary_Elements.Node -->
<!-- _____ -->

<!ELEMENT UML:Node.component (UML:Component)* >

<!ELEMENT UML:Node (UML:ModelElement.name | UML:ModelElement.visibility |
    UML:GeneralizableElement.isRoot |
    UML:GeneralizableElement.isLeaf |
    UML:GeneralizableElement.isAbstract | XMI.extension |
    UML:ModelElement.binding | UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:GeneralizableElement.generalization |
    UML:GeneralizableElement.specialization |
    UML:Classifier.parameter |
    UML:Classifier.structuralFeature |

```

```

        UML:Classifier.specification |
        UML:Classifier.realization |
        UML:Classifier.associationEnd |
        UML:Classifier.participant |
        UML:Classifier.createAction |
        UML:Classifier.instance |
        UML:Classifier.collaboration |
        UML:Classifier.classifierRole |
        UML:Classifier.classifierInState |
        UML:Node.component | UML:ModelElement.taggedValue |
        UML:Namespace.ownedElement |
        UML:Classifier.feature)* >
<!-- UML:Node
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    ModelElement.collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    parameter IDREFS #IMPLIED
    structuralFeature IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    realization IDREFS #IMPLIED
    associationEnd IDREFS #IMPLIED
    participant IDREFS #IMPLIED
    createAction IDREFS #IMPLIED
    instance IDREFS #IMPLIED
    Classifier.collaboration IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    component IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>
<!-- ----- -->
<!-- ----- -->
<!-- CLASS: Foundation.Auxiliary_Elements.Component -->
<!-- ----- -->

```

```

<!ELEMENT UML:Component.deployment (UML:Node)* >

<!ELEMENT UML:Component.implements (UML:ModelElement)* >

<!ELEMENT UML:Component (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:GeneralizableElement.isRoot |
                        UML:GeneralizableElement.isLeaf |
                        UML:GeneralizableElement.isAbstract |
                        XMI.extension | UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:GeneralizableElement.generalization |
                        UML:GeneralizableElement.specialization |
                        UML:Classifier.parameter |
                        UML:Classifier.structuralFeature |
                        UML:Classifier.specification |
                        UML:Classifier.realization |
                        UML:Classifier.associationEnd |
                        UML:Classifier.participant |
                        UML:Classifier.createAction |
                        UML:Classifier.instance |
                        UML:Classifier.collaboration |
                        UML:Classifier.classifierRole |
                        UML:Classifier.classifierInState |
                        UML:Component.deployment |
                        UML:Component.implements |
                        UML:ModelElement.taggedValue |
                        UML:Namespace.ownedElement |
                        UML:Classifier.feature)* >

<!ATTLIST UML:Component
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED

```

```

namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
ModelElement.collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
generalization IDREFS #IMPLIED
specialization IDREFS #IMPLIED
parameter IDREFS #IMPLIED
structuralFeature IDREFS #IMPLIED
specification IDREFS #IMPLIED
realization IDREFS #IMPLIED
associationEnd IDREFS #IMPLIED
participant IDREFS #IMPLIED
createAction IDREFS #IMPLIED
instance IDREFS #IMPLIED
Classifier.collaboration IDREFS #IMPLIED
classifierRole IDREFS #IMPLIED
classifierInState IDREFS #IMPLIED
deployment IDREFS #IMPLIED
implements IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Auxiliary_Elements.Comment -->
<!-- _____ -->

<!ELEMENT UML:Comment (UML:ModelElement.name |
                        UML:ModelElement.visibility | XMI:extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:ModelElement.taggedValue)* >

<!ATTLIST UML:Comment
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED

```

```

        template IDREFS #IMPLIED
        templateParameter IDREFS #IMPLIED
        implementation IDREFS #IMPLIED
        view IDREFS #IMPLIED
        presentation IDREFS #IMPLIED
        namespace IDREFS #IMPLIED
        constraint IDREFS #IMPLIED
        requirement IDREFS #IMPLIED
        provision IDREFS #IMPLIED
        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Auxiliary_Elements.ViewElement -->
<!-- _____ -->

<!ELEMENT UML:ViewElement.model (UML:ModelElement)* >

<!ELEMENT UML:ViewElement.presentation (UML:Presentation)* >

<!ELEMENT UML:ViewElement (XMI.extension | UML:ViewElement.model |
                           UML:ViewElement.presentation)* >
<!ATTLIST UML:ViewElement
        model IDREFS #IMPLIED
        presentation IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Auxiliary_Elements.Presentation -->
<!-- _____ -->

<!ELEMENT UML:Presentation.geometry (UML:Geometry) >

<!ELEMENT UML:Presentation.style (UML:GraphicMarker) >

<!ELEMENT UML:Presentation.model (UML:ModelElement)* >

<!ELEMENT UML:Presentation.viewElement (UML:ViewElement)* >

<!ELEMENT UML:Presentation (UML:Presentation.geometry |
                           UML:Presentation.style | XMI.extension |
                           UML:Presentation.model |
                           UML:Presentation.viewElement)* >
<!ATTLIST UML:Presentation
        model IDREFS #IMPLIED

```

```

        viewElement IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.Element -->
<!-- _____ -->

<!ELEMENT UML:Element (XMI.extension)* >
<!ATTLIST UML:Element
        %XMI.element.att;
        %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.ModelElement -->
<!-- _____ -->

<!ELEMENT UML:ModelElement.name (#PCDATA | XMI.reference)* >

<!ELEMENT UML:ModelElement.visibility EMPTY >
<!ATTLIST UML:ModelElement.visibility
        xmi.value (public | protected | private) #REQUIRED
>

<!ELEMENT UML:ModelElement.binding (UML:Binding)* >

<!ELEMENT UML:ModelElement.template (UML:ModelElement)* >

<!ELEMENT UML:ModelElement.templateParameter (UML:ModelElement)* >

<!ELEMENT UML:ModelElement.implementation (UML:Component)* >

<!ELEMENT UML:ModelElement.view (UML:ViewElement)* >

<!ELEMENT UML:ModelElement.presentation (UML:Presentation)* >

<!ELEMENT UML:ModelElement.namespace (UML:Namespace)* >

<!ELEMENT UML:ModelElement.constraint (UML:Constraint)* >

<!ELEMENT UML:ModelElement.requirement (UML:Dependency)* >

<!ELEMENT UML:ModelElement.provision (UML:Dependency)* >

<!ELEMENT UML:ModelElement.stereotype (UML:Stereotype)* >

<!ELEMENT UML:ModelElement.elementReference (UML:ElementReference)* >

<!ELEMENT UML:ModelElement.collaboration (UML:Collaboration)* >

<!ELEMENT UML:ModelElement.behavior (UML:StateMachine)* >

```

```

<!ELEMENT UML:ModelElement.partition (UML:Partition)* >

<!ELEMENT UML:ModelElement (UML:ModelElement.name |
    UML:ModelElement.visibility | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:ModelElement.taggedValue)* >

<!-- UML:ModelElement
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
-->

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.Namespace -->
<!-- _____ -->

<!ELEMENT UML:Namespace (UML:ModelElement.name |
    UML:ModelElement.visibility | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |

```

```

        UML:ModelElement.presentation |
        UML:ModelElement.namespace |
        UML:ModelElement.constraint |
        UML:ModelElement.requirement |
        UML:ModelElement.provision |
        UML:ModelElement.stereotype |
        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition |
        UML:ModelElement.taggedValue |
        UML:Namespace.ownedElement)* >

<!-- UML:Namespace
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
-->

<!-- CLASS: Foundation.Core.GeneralizableElement -->
<!-- -->

<!-- UML:GeneralizableElement.isRoot EMPTY >
<!-- UML:GeneralizableElement.isRoot
    xmi.value (false | true) #REQUIRED
-->

<!-- UML:GeneralizableElement.isLeaf EMPTY >
<!-- UML:GeneralizableElement.isLeaf
    xmi.value (false | true) #REQUIRED
-->

<!-- UML:GeneralizableElement.isAbstract EMPTY >
<!-- UML:GeneralizableElement.isAbstract
    xmi.value (false | true) #REQUIRED
-->

<!-- UML:GeneralizableElement.generalization

```



```

(UML:Generalization)* >

<!ELEMENT UML:GeneralizableElement.specialization
(UML:Generalization)* >

<!ELEMENT UML:GeneralizableElement (UML:ModelElement.name |
UML:ModelElement.visibility |
UML:GeneralizableElement.isRoot |
UML:GeneralizableElement.isLeaf |
UML:GeneralizableElement.isAbstract |
XMI.extension |
UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:GeneralizableElement.generalization |
UML:GeneralizableElement.specialization |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement)* >

<!ATTLIST UML:GeneralizableElement
name CDATA #IMPLIED
visibility (public | protected | private) #IMPLIED
isRoot (false | true) #IMPLIED
isLeaf (false | true) #IMPLIED
isAbstract (false | true) #IMPLIED
binding IDREFS #IMPLIED
template IDREFS #IMPLIED
templateParameter IDREFS #IMPLIED
implementation IDREFS #IMPLIED
view IDREFS #IMPLIED
presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
generalization IDREFS #IMPLIED
specialization IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;

```

```

>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.Classifier -->
<!-- _____ -->

<!ELEMENT UML:Classifier.parameter (UML:Parameter)* >

<!ELEMENT UML:Classifier.structuralFeature (UML:StructuralFeature)* >

<!ELEMENT UML:Classifier.specification (UML:Classifier)* >

<!ELEMENT UML:Classifier.realization (UML:Classifier)* >

<!ELEMENT UML:Classifier.associationEnd (UML:AssociationEnd)* >

<!ELEMENT UML:Classifier.participant (UML:AssociationEnd)* >

<!ELEMENT UML:Classifier.createAction (UML:CreateAction)* >

<!ELEMENT UML:Classifier.instance (UML:Instance)* >

<!ELEMENT UML:Classifier.collaboration (UML:Collaboration)* >

<!ELEMENT UML:Classifier.classifierRole (UML:ClassifierRole)* >

<!ELEMENT UML:Classifier.classifierInState (UML:ClassifierInState)* >

<!ELEMENT UML:Classifier (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:GeneralizableElement.isRoot |
                        UML:GeneralizableElement.isLeaf |
                        UML:GeneralizableElement.isAbstract |
                        XMI.extension | UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:GeneralizableElement.generalization |
                        UML:GeneralizableElement.specialization |
                        UML:Classifier.parameter |
                        UML:Classifier.structuralFeature |
                        UML:Classifier.specification |
                        UML:Classifier.realization |

```

```

UML:Classifier.associationEnd |
UML:Classifier.participant |
UML:Classifier.createAction |
UML:Classifier.instance |
UML:Classifier.collaboration |
UML:Classifier.classifierRole |
UML:Classifier.classifierInState |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Classifier.feature)* >

<!-- UML:Classifier
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  isRoot (false | true) #IMPLIED
  isLeaf (false | true) #IMPLIED
  isAbstract (false | true) #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  ModelElement.collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED
  partition IDREFS #IMPLIED
  generalization IDREFS #IMPLIED
  specialization IDREFS #IMPLIED
  parameter IDREFS #IMPLIED
  structuralFeature IDREFS #IMPLIED
  specification IDREFS #IMPLIED
  realization IDREFS #IMPLIED
  associationEnd IDREFS #IMPLIED
  participant IDREFS #IMPLIED
  createAction IDREFS #IMPLIED
  instance IDREFS #IMPLIED
  Classifier.collaboration IDREFS #IMPLIED
  classifierRole IDREFS #IMPLIED
  classifierInState IDREFS #IMPLIED
  %XMI.element.att;
  %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.Interface -->
<!-- _____ -->

<!-- UML:Interface (UML:ModelElement.name |
                    UML:ModelElement.visibility |

```

```

UML:GeneralizableElement.isRoot |
UML:GeneralizableElement.isLeaf |
UML:GeneralizableElement.isAbstract |
XMI.extension | UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:GeneralizableElement.generalization |
UML:GeneralizableElement.specialization |
UML:Classifier.parameter |
UML:Classifier.structuralFeature |
UML:Classifier.specification |
UML:Classifier.realization |
UML:Classifier.associationEnd |
UML:Classifier.participant |
UML:Classifier.createAction |
UML:Classifier.instance |
UML:Classifier.collaboration |
UML:Classifier.classifierRole |
UML:Classifier.classifierInState |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Classifier.feature)* >
<!ATTLIST UML:Interface
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    ModelElement.collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED

```

```

        generalization IDREFS #IMPLIED
        specialization IDREFS #IMPLIED
        parameter IDREFS #IMPLIED
        structuralFeature IDREFS #IMPLIED
        specification IDREFS #IMPLIED
        realization IDREFS #IMPLIED
        associationEnd IDREFS #IMPLIED
        participant IDREFS #IMPLIED
        createAction IDREFS #IMPLIED
        instance IDREFS #IMPLIED
        Classifier.collaboration IDREFS #IMPLIED
        classifierRole IDREFS #IMPLIED
        classifierInState IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.Class -->
<!-- _____ -->

<!ELEMENT UML:Class.isActive EMPTY >
<!ATTLIST UML:Class.isActive
            xmi.value (false | true) #REQUIRED
>

<!ELEMENT UML:Class (UML:ModelElement.name |
                    UML:ModelElement.visibility |
                    UML:GeneralizableElement.isRoot |
                    UML:GeneralizableElement.isLeaf |
                    UML:GeneralizableElement.isAbstract |
                    UML:Class.isActive | XMI:extension |
                    UML:ModelElement.binding |
                    UML:ModelElement.template |
                    UML:ModelElement.templateParameter |
                    UML:ModelElement.implementation |
                    UML:ModelElement.view |
                    UML:ModelElement.presentation |
                    UML:ModelElement.namespace |
                    UML:ModelElement.constraint |
                    UML:ModelElement.requirement |
                    UML:ModelElement.provision |
                    UML:ModelElement.stereotype |
                    UML:ModelElement.elementReference |
                    UML:ModelElement.collaboration |
                    UML:ModelElement.behavior |
                    UML:ModelElement.partition |
                    UML:GeneralizableElement.generalization |
                    UML:GeneralizableElement.specialization |
                    UML:Classifier.parameter |
                    UML:Classifier.structuralFeature |
                    UML:Classifier.specification |
                    UML:Classifier.realization |
                    UML:Classifier.associationEnd |

```

```

        UML:Classifier.participant |
        UML:Classifier.createAction |
        UML:Classifier.instance |
        UML:Classifier.collaboration |
        UML:Classifier.classifierRole |
        UML:Classifier.classifierInState |
        UML:ModelElement.taggedValue |
        UML:Namespace.ownedElement |
        UML:Classifier.feature)* >
<!-- UML:Class
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    isActive (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    ModelElement.collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    parameter IDREFS #IMPLIED
    structuralFeature IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    realization IDREFS #IMPLIED
    associationEnd IDREFS #IMPLIED
    participant IDREFS #IMPLIED
    createAction IDREFS #IMPLIED
    instance IDREFS #IMPLIED
    Classifier.collaboration IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
-->

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.DataType -->
<!-- _____ -->

<!-- UML:DataType (UML:ModelElement.name |
                    UML:ModelElement.visibility |

```

```

UML:GeneralizableElement.isRoot |
UML:GeneralizableElement.isLeaf |
UML:GeneralizableElement.isAbstract |
XMI.extension | UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:GeneralizableElement.generalization |
UML:GeneralizableElement.specialization |
UML:Classifier.parameter |
UML:Classifier.structuralFeature |
UML:Classifier.specification |
UML:Classifier.realization |
UML:Classifier.associationEnd |
UML:Classifier.participant |
UML:Classifier.createAction |
UML:Classifier.instance |
UML:Classifier.collaboration |
UML:Classifier.classifierRole |
UML:Classifier.classifierInState |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Classifier.feature)* >
<!ATTLIST UML:DataType
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  isRoot (false | true) #IMPLIED
  isLeaf (false | true) #IMPLIED
  isAbstract (false | true) #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  ModelElement.collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED
  partition IDREFS #IMPLIED

```

```

        generalization IDREFS #IMPLIED
        specialization IDREFS #IMPLIED
        parameter IDREFS #IMPLIED
        structuralFeature IDREFS #IMPLIED
        specification IDREFS #IMPLIED
        realization IDREFS #IMPLIED
        associationEnd IDREFS #IMPLIED
        participant IDREFS #IMPLIED
        createAction IDREFS #IMPLIED
        instance IDREFS #IMPLIED
        Classifier.collaboration IDREFS #IMPLIED
        classifierRole IDREFS #IMPLIED
        classifierInState IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.Feature -->
<!-- _____ -->

<!ELEMENT UML:Feature.ownerScope EMPTY >
<!ATTLIST UML:Feature.ownerScope
            xmi.value (classifier | instance) #REQUIRED
>

<!ELEMENT UML:Feature.owner (UML:Classifier)* >

<!ELEMENT UML:Feature.classifierRole (UML:ClassifierRole)* >

<!ELEMENT UML:Feature (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:Feature.ownerScope | XMI.extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition | UML:Feature.owner |
                        UML:Feature.classifierRole |
                        UML:ModelElement.taggedValue)* >
<!ATTLIST UML:Feature
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    ownerScope (classifier | instance) #IMPLIED

```



```

binding IDREFS #IMPLIED
template IDREFS #IMPLIED
templateParameter IDREFS #IMPLIED
implementation IDREFS #IMPLIED
view IDREFS #IMPLIED
presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
owner IDREFS #IMPLIED
classifierRole IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.StructuralFeature -->
<!-- _____ -->

<!ELEMENT UML:StructuralFeature.multiplicity (#PCDATA |
                                                XMI.reference)* >

<!ELEMENT UML:StructuralFeature.changeable EMPTY >
<!ATTLIST UML:StructuralFeature.changeable
          xmi.value (none | frozen | addOnly) #REQUIRED
>

<!ELEMENT UML:StructuralFeature.targetScope EMPTY >
<!ATTLIST UML:StructuralFeature.targetScope
          xmi.value (classifier | instance) #REQUIRED
>

<!ELEMENT UML:StructuralFeature.type (UML:Classifier)* >

<!ELEMENT UML:StructuralFeature (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:Feature.ownerScope |
                                UML:StructuralFeature.multiplicity |
                                UML:StructuralFeature.changeable |
                                UML:StructuralFeature.targetScope |
                                XMI.extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |

```

```

UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:Feature.owner |
UML:Feature.classifierRole |
UML:StructuralFeature.type |
UML:ModelElement.taggedValue)* >

<!-- UML:StructuralFeature
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  ownerScope (classifier | instance) #IMPLIED
  multiplicity CDATA #IMPLIED
  changeable (none | frozen | addOnly) #IMPLIED
  targetScope (classifier | instance) #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED
  partition IDREFS #IMPLIED
  owner IDREFS #IMPLIED
  classifierRole IDREFS #IMPLIED
  type IDREFS #IMPLIED
  %XMI.element.att;
  %XMI.link.att;
-->

<!-- CLASS: Foundation.Core.BehavioralFeature
-->

<!-- UML:BehavioralFeature.isQuery EMPTY >
<!-- UML:BehavioralFeature.isQuery
      xmi.value (false | true) #REQUIRED
-->

<!-- UML:BehavioralFeature.raisedException (UML:Exception)* >

<!-- UML:BehavioralFeature (UML:ModelElement.name |
                          UML:ModelElement.visibility |

```

```

UML:Feature.ownerScope |
UML:BehavioralFeature.isQuery |
XMI.extension |
UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:Feature.owner |
UML:Feature.classifierRole |
UML:BehavioralFeature.raisedException |
UML:ModelElement.taggedValue |
UML:BehavioralFeature.parameter)* >

<!-- UML:BehavioralFeature
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    ownerScope (classifier | instance) #IMPLIED
    isQuery (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owner IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    raisedException IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.Operation -->
<!-- _____ -->

```

```

<!ELEMENT UML:Operation.specification (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Operation.isPolymorphic EMPTY >
<!ATTLIST UML:Operation.isPolymorphic
          xmi.value (false | true) #REQUIRED
>

<!ELEMENT UML:Operation.concurrency EMPTY >
<!ATTLIST UML:Operation.concurrency
          xmi.value (sequential | guarded | concurrent) #REQUIRED
>

<!ELEMENT UML:Operation.method (UML:Method)* >

<!ELEMENT UML:Operation.collaboration (UML:Collaboration)* >

<!ELEMENT UML:Operation.occurrence (UML:CallEvent)* >

<!ELEMENT UML:Operation (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:Feature.ownerScope |
                        UML:BehavioralFeature.isQuery |
                        UML:Operation.specification |
                        UML:Operation.isPolymorphic |
                        UML:Operation.concurrency | XMI.extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition | UML:Feature.owner |
                        UML:Feature.classifierRole |
                        UML:BehavioralFeature.raisedException |
                        UML:Request.action |
                        UML:Request.messageInstance |
                        UML:Operation.method |
                        UML:Operation.collaboration |
                        UML:Operation.occurrence |
                        UML:ModelElement.taggedValue |
                        UML:BehavioralFeature.parameter)* >

<!ATTLIST UML:Operation
          name CDATA #IMPLIED
          visibility (public | protected | private) #IMPLIED
          ownerScope (classifier | instance) #IMPLIED
          isQuery (false | true) #IMPLIED
          specification CDATA #IMPLIED

```

```

isPolymorphic (false | true) #IMPLIED
concurrency (sequential | guarded | concurrent) #IMPLIED
binding IDREFS #IMPLIED
template IDREFS #IMPLIED
templateParameter IDREFS #IMPLIED
implementation IDREFS #IMPLIED
view IDREFS #IMPLIED
presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
ModelElement.collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
owner IDREFS #IMPLIED
classifierRole IDREFS #IMPLIED
raisedException IDREFS #IMPLIED
action IDREFS #IMPLIED
messageInstance IDREFS #IMPLIED
method IDREFS #IMPLIED
Operation.collaboration IDREFS #IMPLIED
occurrence IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.Method -->
<!-- _____ -->

<!ELEMENT UML:Method.body (UML:ProcedureExpression) >

<!ELEMENT UML:Method.specification (UML:Operation)* >

<!ELEMENT UML:Method (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:Feature.ownerScope |
    UML:BehavioralFeature.isQuery | UML:Method.body |
    XMI.extension | UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |

```

```

        UML:ModelElement.behavior |
        UML:ModelElement.partition | UML:Feature.owner |
        UML:Feature.classifierRole |
        UML:BehavioralFeature.raisedException |
        UML:Method.specification |
        UML:ModelElement.taggedValue |
        UML:BehavioralFeature.parameter)* >
<!-- UML:Method
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    ownerScope (classifier | instance) #IMPLIED
    isQuery (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owner IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    raisedException IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.Parameter -->
<!-- _____ -->

<!-- UML:Parameter.defaultValue (UML:Expression |
                                UML:ProcedureExpression |
                                UML:ObjectSetExpression |
                                UML:TimeExpression |
                                UML:BooleanExpression) >

<!-- UML:Parameter.kind EMPTY >
<!-- UML:Parameter.kind
    xmi.value (in | inout | out | return) #REQUIRED
>

<!-- UML:Parameter.behavioralFeature (UML:BehavioralFeature)* >

<!-- UML:Parameter.type (UML:Classifier)* >

```

```

<!ELEMENT UML:Parameter.signal (UML:Signal)* >

<!ELEMENT UML:Parameter (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:Parameter.defaultValue | UML:Parameter.kind |
    XMI.extension | UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:Parameter.behavioralFeature |
    UML:Parameter.type | UML:Parameter.signal |
    UML:ModelElement.taggedValue)* >

<!ATTLIST UML:Parameter
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    kind (in | inout | out | return) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    behavioralFeature IDREFS #IMPLIED
    type IDREFS #IMPLIED
    signal IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;

>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.Constraint -->
<!-- _____ -->

<!ELEMENT UML:Constraint.body (UML:BooleanExpression) >

```

```

<!ELEMENT UML:Constraint.constrainedElement (UML:ModelElement)* >

<!ELEMENT UML:Constraint.constrainedStereotype (UML:Stereotype)* >

<!ELEMENT UML:Constraint (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:Constraint.body | XMI.extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:Constraint.constrainedElement |
                        UML:Constraint.constrainedStereotype |
                        UML:ModelElement.taggedValue)* >

<!ATTLIST UML:Constraint
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    constrainedElement IDREFS #IMPLIED
    constrainedStereotype IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.Dependency -->
<!-- _____ -->

```



```

<!ELEMENT UML:Dependency.description (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Dependency.owningDependency (UML:Dependency)* >

<!ELEMENT UML:Dependency.client (UML:ModelElement)* >

<!ELEMENT UML:Dependency.supplier (UML:ModelElement)* >

<!ELEMENT UML:Dependency (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:Dependency.description | XMI.extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:Dependency.owningDependency |
                        UML:Dependency.client |
                        UML:Dependency.supplier |
                        UML:ModelElement.taggedValue |
                        UML:Dependency.subDependencies)* >

<!ATTLIST UML:Dependency
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    description CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owningDependency IDREFS #IMPLIED
    client IDREFS #IMPLIED
    supplier IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;

```

```

>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.Generalization -->
<!-- _____ -->

<!ELEMENT UML:Generalization.discriminator (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Generalization.subtype (UML:GeneralizableElement)* >

<!ELEMENT UML:Generalization.supertype (UML:GeneralizableElement)* >

<!ELEMENT UML:Generalization (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:Generalization.discriminator |
                                XMI.extension | UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:Generalization.subtype |
                                UML:Generalization.supertype |
                                UML:ModelElement.taggedValue)* >

<!-- ATTLIST UML:Generalization
name CDATA #IMPLIED
visibility (public | protected | private) #IMPLIED
discriminator CDATA #IMPLIED
binding IDREFS #IMPLIED
template IDREFS #IMPLIED
templateParameter IDREFS #IMPLIED
implementation IDREFS #IMPLIED
view IDREFS #IMPLIED
presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
subtype IDREFS #IMPLIED
supertype IDREFS #IMPLIED
-->

```

```

        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.AssociationEnd -->
<!-- _____ -->

<!ELEMENT UML:AssociationEnd.isNavigable EMPTY >
<!ATTLIST UML:AssociationEnd.isNavigable
            xmi.value (false | true) #REQUIRED
>

<!ELEMENT UML:AssociationEnd.isOrdered EMPTY >
<!ATTLIST UML:AssociationEnd.isOrdered
            xmi.value (false | true) #REQUIRED
>

<!ELEMENT UML:AssociationEnd.aggregation EMPTY >
<!ATTLIST UML:AssociationEnd.aggregation
            xmi.value (none | shared | composite) #REQUIRED
>

<!ELEMENT UML:AssociationEnd.multiplicity (#PCDATA | XMI:reference)* >

<!ELEMENT UML:AssociationEnd.changeable EMPTY >
<!ATTLIST UML:AssociationEnd.changeable
            xmi.value (none | frozen | addOnly) #REQUIRED
>

<!ELEMENT UML:AssociationEnd.targetScope EMPTY >
<!ATTLIST UML:AssociationEnd.targetScope
            xmi.value (classifier | instance) #REQUIRED
>

<!ELEMENT UML:AssociationEnd.type (UML:Classifier)* >

<!ELEMENT UML:AssociationEnd.specification (UML:Classifier)* >

<!ELEMENT UML:AssociationEnd.association (UML:Association)* >

<!ELEMENT UML:AssociationEnd.linkEnd (UML:LinkEnd)* >

<!ELEMENT UML:AssociationEnd.associationEndRole
                                            (UML:AssociationEndRole)* >

<!ELEMENT UML:AssociationEnd (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:AssociationEnd.isNavigable |
                                UML:AssociationEnd.isOrdered |
                                UML:AssociationEnd.aggregation |
                                UML:AssociationEnd.multiplicity |
                                UML:AssociationEnd.changeable |
                                UML:AssociationEnd.targetScope |

```

```

XMI.extension | UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:AssociationEnd.type |
UML:AssociationEnd.specification |
UML:AssociationEnd.association |
UML:AssociationEnd.linkEnd |
UML:AssociationEnd.associationEndRole |
UML:ModelElement.taggedValue |
UML:AssociationEnd.qualifier)* >

<!ATTLIST UML:AssociationEnd
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isNavigable (false | true) #IMPLIED
    isOrdered (false | true) #IMPLIED
    aggregation (none | shared | composite) #IMPLIED
    multiplicity CDATA #IMPLIED
    changeable (none | frozen | addOnly) #IMPLIED
    targetScope (classifier | instance) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    type IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    association IDREFS #IMPLIED
    linkEnd IDREFS #IMPLIED
    associationEndRole IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

```

```

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Core.Association -->
<!-- _____ -->

<!ELEMENT UML:Association.link (UML:Link)* >

<!ELEMENT UML:Association.associationEnd (UML:AssociationRole)* >

<!ELEMENT UML:Association (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:GeneralizableElement.isRoot |
    UML:GeneralizableElement.isLeaf |
    UML:GeneralizableElement.isAbstract |
    XML:extension | UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:GeneralizableElement.generalization |
    UML:GeneralizableElement.specialization |
    UML:Association.link |
    UML:Association.associationEnd |
    UML:ModelElement.taggedValue |
    UML:Namespace.ownedElement |
    UML:Association.connection)* >

<!ATTLIST UML:Association
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED

```

```

behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
generalization IDREFS #IMPLIED
specialization IDREFS #IMPLIED
link IDREFS #IMPLIED
associationEnd IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.AssociationClass -->
<!-- _____ -->

<!ELEMENT UML:AssociationClass (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:GeneralizableElement.isRoot |
                                UML:GeneralizableElement.isLeaf |
                                UML:GeneralizableElement.isAbstract |
                                UML:Class.isActive | XMI.extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:GeneralizableElement.generalization |
                                UML:GeneralizableElement.specialization |
                                UML:Classifier.parameter |
                                UML:Classifier.structuralFeature |
                                UML:Classifier.specification |
                                UML:Classifier.realization |
                                UML:Classifier.associationEnd |
                                UML:Classifier.participant |
                                UML:Classifier.createAction |
                                UML:Classifier.instance |
                                UML:Classifier.collaboration |
                                UML:Classifier.classifierRole |
                                UML:Classifier.classifierInState |
                                UML:Association.link |
                                UML:Association.associationEnd |
                                UML:ModelElement.taggedValue |
                                UML:Namespace.ownedElement |
                                UML:Classifier.feature |
                                UML:Association.connection)* >

```

```

<!ATTLIST UML:AssociationClass
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    isActive (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    ModelElement.collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    parameter IDREFS #IMPLIED
    structuralFeature IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    realization IDREFS #IMPLIED
    Classifier.associationEnd IDREFS #IMPLIED
    participant IDREFS #IMPLIED
    createAction IDREFS #IMPLIED
    instance IDREFS #IMPLIED
    Classifier.collaboration IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    link IDREFS #IMPLIED
    Association.associationEnd IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Core.Attribute -->
<!-- _____ -->

<!ELEMENT UML:Attribute.initialValue (UML:Expression |
                                         UML:ProcedureExpression |
                                         UML:ObjectSetExpression |
                                         UML:TimeExpression |
                                         UML:BooleanExpression) >

<!ELEMENT UML:Attribute.associationEnd (UML:AssociationEnd)* >

<!ELEMENT UML:Attribute.attributeLink (UML:AttributeLink)* >

```

```

<!ELEMENT UML:Attribute (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:Feature.ownerScope |
                        UML:StructuralFeature.multiplicity |
                        UML:StructuralFeature.changeable |
                        UML:StructuralFeature.targetScope |
                        UML:Attribute.initialValue | XMI.extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition | UML:Feature.owner |
                        UML:Feature.classifierRole |
                        UML:StructuralFeature.type |
                        UML:Attribute.associationEnd |
                        UML:Attribute.attributeLink |
                        UML:ModelElement.taggedValue)* >

<!ATTLIST UML:Attribute
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    ownerScope (classifier | instance) #IMPLIED
    multiplicity CDATA #IMPLIED
    changeable (none | frozen | addOnly) #IMPLIED
    targetScope (classifier | instance) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owner IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    type IDREFS #IMPLIED
    associationEnd IDREFS #IMPLIED
    attributeLink IDREFS #IMPLIED

```



```

        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Extension_Mechanisms.TaggedValue -->
<!-- _____ -->

<!ELEMENT UML:TaggedValue.tag (#PCDATA | XMI.reference)* >

<!ELEMENT UML:TaggedValue.value (#PCDATA | XMI.reference)* >

<!ELEMENT UML:TaggedValue.modelElement (UML:ModelElement)* >

<!ELEMENT UML:TaggedValue.stereotype (UML:Stereotype)* >

<!ELEMENT UML:TaggedValue (UML:TaggedValue.tag | UML:TaggedValue.value |
                        XMI.extension | UML:TaggedValue.modelElement |
                        UML:TaggedValue.stereotype)* >

<!ATTLIST UML:TaggedValue
    tag CDATA #IMPLIED
    value CDATA #IMPLIED
    modelElement IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Extension_Mechanisms.Stereotype -->
<!-- _____ -->

<!ELEMENT UML:Stereotype.icon (UML:Geometry) >

<!ELEMENT UML:Stereotype.baseClass (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Stereotype.extendedElement (UML:ModelElement)* >

<!ELEMENT UML:Stereotype.stereotypeConstraint (UML:Constraint)* >

<!ELEMENT UML:Stereotype (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:GeneralizableElement.isRoot |
                        UML:GeneralizableElement.isLeaf |
                        UML:GeneralizableElement.isAbstract |
                        UML:Stereotype.icon | UML:Stereotype.baseClass |
                        XMI.extension | UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |

```

```

UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:GeneralizableElement.generalization |
UML:GeneralizableElement.specialization |
UML:Stereotype.extendedElement |
UML:Stereotype.stereotypeConstraint |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Stereotype.requiredTag)* >

<!-- UML:Stereotype
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  isRoot (false | true) #IMPLIED
  isLeaf (false | true) #IMPLIED
  isAbstract (false | true) #IMPLIED
  baseClass CDATA #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED
  partition IDREFS #IMPLIED
  generalization IDREFS #IMPLIED
  specialization IDREFS #IMPLIED
  extendedElement IDREFS #IMPLIED
  stereotypeConstraint IDREFS #IMPLIED
  %XMI.element.att;
  %XMI.link.att;
-->

<!-- CLASS: Foundation.Data_Types.Enumeration
-->

<!-- UML:Enumeration (UML:ModelElement.name |
  UML:ModelElement.visibility |
  UML:GeneralizableElement.isRoot |
  UML:GeneralizableElement.isLeaf |
  UML:GeneralizableElement.isAbstract |

```

```

XMI:extension | UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:GeneralizableElement.generalization |
UML:GeneralizableElement.specialization |
UML:Classifier.parameter |
UML:Classifier.structuralFeature |
UML:Classifier.specification |
UML:Classifier.realization |
UML:Classifier.associationEnd |
UML:Classifier.participant |
UML:Classifier.createAction |
UML:Classifier.instance |
UML:Classifier.collaboration |
UML:Classifier.classifierRole |
UML:Classifier.classifierInState |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Classifier.feature |
UML:Enumeration.literal)* >
<!ATTLIST UML:Enumeration
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    ModelElement.collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED

```

```

        parameter IDREFS #IMPLIED
        structuralFeature IDREFS #IMPLIED
        specification IDREFS #IMPLIED
        realization IDREFS #IMPLIED
        associationEnd IDREFS #IMPLIED
        participant IDREFS #IMPLIED
        createAction IDREFS #IMPLIED
        instance IDREFS #IMPLIED
        Classifier.collaboration IDREFS #IMPLIED
        classifierRole IDREFS #IMPLIED
        classifierInState IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Data_Types.EnumerationLiteral -->
<!-- _____ -->

<!ELEMENT UML:EnumerationLiteral.name (#PCDATA | XMI.reference)* >

<!ELEMENT UML:EnumerationLiteral.enumeration (UML:Enumeration)* >

<!ELEMENT UML:EnumerationLiteral (UML:EnumerationLiteral.name |
                                   XMI.extension |
                                   UML:EnumerationLiteral.enumeration)* >

<ATTLIST UML:EnumerationLiteral
    name CDATA #IMPLIED
    enumeration IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Data_Types.Primitive -->
<!-- _____ -->

<!ELEMENT UML:Primitive (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:GeneralizableElement.isRoot |
                        UML:GeneralizableElement.isLeaf |
                        UML:GeneralizableElement.isAbstract |
                        XMI.extension | UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |

```

```

        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition |
        UML:GeneralizableElement.generalization |
        UML:GeneralizableElement.specialization |
        UML:Classifier.parameter |
        UML:Classifier.structuralFeature |
        UML:Classifier.specification |
        UML:Classifier.realization |
        UML:Classifier.associationEnd |
        UML:Classifier.participant |
        UML:Classifier.createAction |
        UML:Classifier.instance |
        UML:Classifier.collaboration |
        UML:Classifier.classifierRole |
        UML:Classifier.classifierInState |
        UML:ModelElement.taggedValue |
        UML:Namespace.ownedElement |
        UML:Classifier.feature)* >
<!ATTLIST UML:Primitive
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    ModelElement.collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    parameter IDREFS #IMPLIED
    structuralFeature IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    realization IDREFS #IMPLIED
    associationEnd IDREFS #IMPLIED
    participant IDREFS #IMPLIED
    createAction IDREFS #IMPLIED
    instance IDREFS #IMPLIED
    Classifier.collaboration IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    %XMI.element.att;
```

```

                                %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Foundation.Data_Types.Structure -->
<!-- _____ -->

<!ELEMENT UML:Structure (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:GeneralizableElement.isRoot |
                        UML:GeneralizableElement.isLeaf |
                        UML:GeneralizableElement.isAbstract |
                        XMI.extension | UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:GeneralizableElement.generalization |
                        UML:GeneralizableElement.specialization |
                        UML:Classifier.parameter |
                        UML:Classifier.structuralFeature |
                        UML:Classifier.specification |
                        UML:Classifier.realization |
                        UML:Classifier.associationEnd |
                        UML:Classifier.participant |
                        UML:Classifier.createAction |
                        UML:Classifier.instance |
                        UML:Classifier.collaboration |
                        UML:Classifier.classifierRole |
                        UML:Classifier.classifierInState |
                        UML:ModelElement.taggedValue |
                        UML:Namespace.ownedElement |
                        UML:Classifier.feature)* >

<!ATTLIST UML:Structure
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED

```

```

        presentation IDREFS #IMPLIED
        namespace IDREFS #IMPLIED
        constraint IDREFS #IMPLIED
        requirement IDREFS #IMPLIED
        provision IDREFS #IMPLIED
        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        ModelElement.collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        generalization IDREFS #IMPLIED
        specialization IDREFS #IMPLIED
        parameter IDREFS #IMPLIED
        structuralFeature IDREFS #IMPLIED
        specification IDREFS #IMPLIED
        realization IDREFS #IMPLIED
        associationEnd IDREFS #IMPLIED
        participant IDREFS #IMPLIED
        createAction IDREFS #IMPLIED
        instance IDREFS #IMPLIED
        Classifier.collaboration IDREFS #IMPLIED
        classifierRole IDREFS #IMPLIED
        classifierInState IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Data_Types.MultiplicityRange -->
<!-- _____ -->

<!ELEMENT UML:MultiplicityRange.lower (#PCDATA | XMI.reference)* >

<!ELEMENT UML:MultiplicityRange.upper (#PCDATA | XMI.reference)* >

<!ELEMENT UML:MultiplicityRange (UML:MultiplicityRange.lower |
                                UML:MultiplicityRange.upper |
                                XMI.extension)* >

<!ATTLIST UML:MultiplicityRange
    lower CDATA #IMPLIED
    upper CDATA #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Data_Types.Geometry -->
<!-- _____ -->

<!ELEMENT UML:Geometry.body (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Geometry (UML:Geometry.body | XMI.extension)* >
<!ATTLIST UML:Geometry

```

```

        body CDATA #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Data_Types.GraphicMarker -->
<!-- _____ -->

<!ELEMENT UML:GraphicMarker.body (#PCDATA | XMI.reference)* >

<!ELEMENT UML:GraphicMarker (UML:GraphicMarker.body | XMI.extension)* >
<!ATTLIST UML:GraphicMarker
    body CDATA #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Data_Types.Mapping -->
<!-- _____ -->

<!ELEMENT UML:Mapping.body (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Mapping (UML:Mapping.body | XMI.extension)* >
<!ATTLIST UML:Mapping
    body CDATA #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Data_Types.Expression -->
<!-- _____ -->

<!ELEMENT UML:Expression.language (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Expression.body (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Expression (UML:Expression.language | UML:Expression.body |
                           XMI.extension)* >
<!ATTLIST UML:Expression
    language CDATA #IMPLIED
    body CDATA #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Foundation.Data_Types.ProcedureExpression -->
<!-- _____ -->

```


10/25/99 *ad/99-10-03: XML Metadata Interchange – Appendices* A-57

```

<!-- _____ -->

<!ELEMENT UML:Package.elementReference (UML:ElementReference)* >

<!ELEMENT UML:Package (UML:ModelElement.name |
                        UML:ModelElement.visibility |
                        UML:GeneralizableElement.isRoot |
                        UML:GeneralizableElement.isLeaf |
                        UML:GeneralizableElement.isAbstract |
                        XMI.extension | UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:GeneralizableElement.generalization |
                        UML:GeneralizableElement.specialization |
                        UML:Package.elementReference |
                        UML:ModelElement.taggedValue |
                        UML:Namespace.ownedElement)* >

<!ATTLIST UML:Package
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    ModelElement.elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    Package.elementReference IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;

```

```

>

<!-- _____ -->
<!-- -->
<!-- CLASS: Model_Management.Subsystem -->
<!-- _____ -->

<!ELEMENT UML:Subsystem.isInstantiable EMPTY >
<!ATTLIST UML:Subsystem.isInstantiable
          xmi.value (false | true) #REQUIRED
>

<!ELEMENT UML:Subsystem (UML:ModelElement.name |
                          UML:ModelElement.visibility |
                          UML:GeneralizableElement.isRoot |
                          UML:GeneralizableElement.isLeaf |
                          UML:GeneralizableElement.isAbstract |
                          UML:Subsystem.isInstantiable | XMI.extension |
                          UML:ModelElement.binding |
                          UML:ModelElement.template |
                          UML:ModelElement.templateParameter |
                          UML:ModelElement.implementation |
                          UML:ModelElement.view |
                          UML:ModelElement.presentation |
                          UML:ModelElement.namespace |
                          UML:ModelElement.constraint |
                          UML:ModelElement.requirement |
                          UML:ModelElement.provision |
                          UML:ModelElement.stereotype |
                          UML:ModelElement.elementReference |
                          UML:ModelElement.collaboration |
                          UML:ModelElement.behavior |
                          UML:ModelElement.partition |
                          UML:GeneralizableElement.generalization |
                          UML:GeneralizableElement.specialization |
                          UML:Classifier.parameter |
                          UML:Classifier.structuralFeature |
                          UML:Classifier.specification |
                          UML:Classifier.realization |
                          UML:Classifier.associationEnd |
                          UML:Classifier.participant |
                          UML:Classifier.createAction |
                          UML:Classifier.instance |
                          UML:Classifier.collaboration |
                          UML:Classifier.classifierRole |
                          UML:Classifier.classifierInState |
                          UML:Package.elementReference |
                          UML:ModelElement.taggedValue |
                          UML:Namespace.ownedElement |
                          UML:Classifier.feature)* >

<!ATTLIST UML:Subsystem
          name CDATA #IMPLIED
          visibility (public | protected | private) #IMPLIED
          isRoot (false | true) #IMPLIED
          isLeaf (false | true) #IMPLIED

```

```

isAbstract (false | true) #IMPLIED
isInstantiable (false | true) #IMPLIED
binding IDREFS #IMPLIED
template IDREFS #IMPLIED
templateParameter IDREFS #IMPLIED
implementation IDREFS #IMPLIED
view IDREFS #IMPLIED
presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
ModelElement.elementReference IDREFS #IMPLIED
ModelElement.collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
generalization IDREFS #IMPLIED
specialization IDREFS #IMPLIED
parameter IDREFS #IMPLIED
structuralFeature IDREFS #IMPLIED
specification IDREFS #IMPLIED
realization IDREFS #IMPLIED
associationEnd IDREFS #IMPLIED
participant IDREFS #IMPLIED
createAction IDREFS #IMPLIED
instance IDREFS #IMPLIED
Classifier.collaboration IDREFS #IMPLIED
classifierRole IDREFS #IMPLIED
classifierInState IDREFS #IMPLIED
Package.elementReference IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Model_Management.Model -->
<!-- _____ -->

<!ELEMENT UML:Model (UML:ModelElement.name |
                    UML:ModelElement.visibility |
                    UML:GeneralizableElement.isRoot |
                    UML:GeneralizableElement.isLeaf |
                    UML:GeneralizableElement.isAbstract | XMI.extension |
                    UML:ModelElement.binding |
                    UML:ModelElement.template |
                    UML:ModelElement.templateParameter |
                    UML:ModelElement.implementation |
                    UML:ModelElement.view |
                    UML:ModelElement.presentation |
                    UML:ModelElement.namespace |
                    UML:ModelElement.constraint |
                    UML:ModelElement.requirement |
                    UML:ModelElement.provision |

```

```

        UML:ModelElement.stereotype |
        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition |
        UML:GeneralizableElement.generalization |
        UML:GeneralizableElement.specialization |
        UML:Package.elementReference |
        UML:ModelElement.taggedValue |
        UML:Namespace.ownedElement)* >
<!-- UML:Model
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    ModelElement.elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    Package.elementReference IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- -->
<!-- CLASS: Model_Management.ElementReference -->
<!-- _____ -->

<!-- UML:ElementReference.visibility EMPTY >
<!-- UML:ElementReference.visibility
        xmi.value (public | protected | private) #REQUIRED
>

<!-- UML:ElementReference.alias (#PCDATA | XMI.reference)* >

<!-- UML:ElementReference.referencedElement (UML:ModelElement)* >

<!-- UML:ElementReference.package (UML:Package)* >

<!-- UML:ElementReference (UML:ElementReference.visibility |

```

```

                                UML:ElementReference.alias |
                                XMI.extension |
                                UML:ElementReference.referencedElement |
                                UML:ElementReference.package)* >
<!-- UML:ElementReference
    visibility (public | protected | private) #IMPLIED
    alias CDATA #IMPLIED
    referencedElement IDREFS #IMPLIED
    package IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Request -->
<!-- _____ -->

<!-- UML:Request.action (UML:Action)* >

<!-- UML:Request.messageInstance (UML:MessageInstance)* >

<!-- UML:Request (UML:ModelElement.name |
                                UML:ModelElement.visibility | XMI.extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition | UML:Request.action |
                                UML:Request.messageInstance |
                                UML:ModelElement.taggedValue)* >
<!-- UML:Request
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED

```

```

        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        action IDREFS #IMPLIED
        messageInstance IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Signal -->
<!-- _____ -->

<!ELEMENT UML:Signal.reception (UML:Reception)* >

<!ELEMENT UML:Signal.occurrence (UML:SignalEvent)* >

<!ELEMENT UML:Signal (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:GeneralizableElement.isRoot |
    UML:GeneralizableElement.isLeaf |
    UML:GeneralizableElement.isAbstract |
    XMI.extension | UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:GeneralizableElement.generalization |
    UML:GeneralizableElement.specialization |
    UML:Request.action | UML:Request.messageInstance |
    UML:Signal.reception | UML:Signal.occurrence |
    UML:ModelElement.taggedValue |
    UML:Namespace.ownedElement |
    UML:Signal.parameter)* >

<!ATTLIST UML:Signal
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED

```

```

implementation IDREFS #IMPLIED
view IDREFS #IMPLIED
presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
generalization IDREFS #IMPLIED
specialization IDREFS #IMPLIED
action IDREFS #IMPLIED
messageInstance IDREFS #IMPLIED
reception IDREFS #IMPLIED
occurrence IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Exception -->
<!-- _____ -->

<!ELEMENT UML:Exception.context (UML:BehavioralFeature)* >

<!ELEMENT UML:Exception (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:GeneralizableElement.isRoot |
    UML:GeneralizableElement.isLeaf |
    UML:GeneralizableElement.isAbstract |
    XMI.extension | UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:GeneralizableElement.generalization |
    UML:GeneralizableElement.specialization |
    UML:Request.action |
    UML:Request.messageInstance |
    UML:Signal.reception | UML:Signal.occurrence |
    UML:Exception.context |

```



```

                                UML:ModelElement.taggedValue |
                                UML:Namespace.ownedElement |
                                UML:Signal.parameter)* >
<!ATTLIST UML:Exception
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    action IDREFS #IMPLIED
    messageInstance IDREFS #IMPLIED
    reception IDREFS #IMPLIED
    occurrence IDREFS #IMPLIED
    context IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Reception -->
<!-- _____ -->

<!ELEMENT UML:Reception.isPolymorphic EMPTY >
<!ATTLIST UML:Reception.isPolymorphic
    xmi.value (false | true) #REQUIRED
>

<!ELEMENT UML:Reception.specification (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Reception.signal (UML:Signal)* >

<!ELEMENT UML:Reception (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:Feature.ownerScope |
    UML:BehavioralFeature.isQuery |
    UML:Reception.isPolymorphic |
    UML:Reception.specification | XMI.extension |

```

```

        UML:ModelElement.binding |
        UML:ModelElement.template |
        UML:ModelElement.templateParameter |
        UML:ModelElement.implementation |
        UML:ModelElement.view |
        UML:ModelElement.presentation |
        UML:ModelElement.namespace |
        UML:ModelElement.constraint |
        UML:ModelElement.requirement |
        UML:ModelElement.provision |
        UML:ModelElement.stereotype |
        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition | UML:Feature.owner |
        UML:Feature.classifierRole |
        UML:BehavioralFeature.raisedException |
        UML:Reception.signal |
        UML:ModelElement.taggedValue |
        UML:BehavioralFeature.parameter)* >
<!-- UML:Reception
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    ownerScope (classifier | instance) #IMPLIED
    isQuery (false | true) #IMPLIED
    isPolymorphic (false | true) #IMPLIED
    specification CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    owner IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    raisedException IDREFS #IMPLIED
    signal IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Argument -->
<!-- _____ -->

```

```

<!ELEMENT UML:Argument.value (UML:Expression | UML:ProcedureExpression |
                               UML:ObjectSetExpression |
                               UML:TimeExpression |
                               UML:BooleanExpression) >

<!ELEMENT UML:Argument.action (UML:Action)* >

<!ELEMENT UML:Argument (UML:Argument.value | XMI.extension |
                        UML:Argument.action)* >

<!ATTLIST UML:Argument
          action IDREFS #IMPLIED
          %XMI.element.att;
          %XMI.link.att;

>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.ActionSequence -->
<!-- _____ -->

<!ELEMENT UML:ActionSequence.transition (UML:Transition)* >

<!ELEMENT UML:ActionSequence.state (UML:State)* >

<!ELEMENT UML:ActionSequence.state2 (UML:State)* >

<!ELEMENT UML:ActionSequence (UML:ModelElement.name |
                               UML:ModelElement.visibility |
                               XMI.extension | UML:ModelElement.binding |
                               UML:ModelElement.template |
                               UML:ModelElement.templateParameter |
                               UML:ModelElement.implementation |
                               UML:ModelElement.view |
                               UML:ModelElement.presentation |
                               UML:ModelElement.namespace |
                               UML:ModelElement.constraint |
                               UML:ModelElement.requirement |
                               UML:ModelElement.provision |
                               UML:ModelElement.stereotype |
                               UML:ModelElement.elementReference |
                               UML:ModelElement.collaboration |
                               UML:ModelElement.behavior |
                               UML:ModelElement.partition |
                               UML:ActionSequence.transition |
                               UML:ActionSequence.state |
                               UML:ActionSequence.state2 |
                               UML:ModelElement.taggedValue |
                               UML:ActionSequence.action)* >

<!ATTLIST UML:ActionSequence
          name CDATA #IMPLIED
          visibility (public | protected | private) #IMPLIED
          binding IDREFS #IMPLIED
          template IDREFS #IMPLIED
          templateParameter IDREFS #IMPLIED
          implementation IDREFS #IMPLIED

```

```

        view IDREFS #IMPLIED
        presentation IDREFS #IMPLIED
        namespace IDREFS #IMPLIED
        constraint IDREFS #IMPLIED
        requirement IDREFS #IMPLIED
        provision IDREFS #IMPLIED
        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        transition IDREFS #IMPLIED
        state IDREFS #IMPLIED
        state2 IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Action -->
<!-- _____ -->

<!ELEMENT UML:Action.recurrence (UML:Expression |
                                UML:ProcedureExpression |
                                UML:ObjectSetExpression |
                                UML:TimeExpression |
                                UML:BooleanExpression) >

<!ELEMENT UML:Action.target (UML:ObjectSetExpression) >

<!ELEMENT UML:Action.isAsynchronous EMPTY >
<!ATTLIST UML:Action.isAsynchronous
          xmi.value (false | true) #REQUIRED
>

<!ELEMENT UML:Action.script (#PCDATA | XMI.reference)* >

<!ELEMENT UML:Action.request (UML:Request)* >

<!ELEMENT UML:Action.message (UML:Message)* >

<!ELEMENT UML:Action.actionSequence (UML:ActionSequence)* >

<!ELEMENT UML:Action (UML:ModelElement.name |
                     UML:ModelElement.visibility |
                     UML:Action.recurrence | UML:Action.target |
                     UML:Action.isAsynchronous | UML:Action.script |
                     XMI.extension | UML:ModelElement.binding |
                     UML:ModelElement.template |
                     UML:ModelElement.templateParameter |
                     UML:ModelElement.implementation |
                     UML:ModelElement.view |
                     UML:ModelElement.presentation |
                     UML:ModelElement.namespace |

```

```

        UML:ModelElement.constraint |
        UML:ModelElement.requirement |
        UML:ModelElement.provision |
        UML:ModelElement.stereotype |
        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition | UML:Action.request |
        UML:Action.message | UML:Action.actionSequence |
        UML:ModelElement.taggedValue |
        UML:Action.actualArgument)* >

<!-- UML:Action
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isAsynchronous (false | true) #IMPLIED
    script CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    request IDREFS #IMPLIED
    message IDREFS #IMPLIED
    actionSequence IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.CreateAction -->
<!-- _____ -->

<!-- UML:CreateAction.instantiation (UML:Classifier)* >

<!-- UML:CreateAction (UML:ModelElement.name |
        UML:ModelElement.visibility |
        UML:Action.recurrence | UML:Action.target |
        UML:Action.isAsynchronous |
        UML:Action.script | XMI.extension |
        UML:ModelElement.binding |
        UML:ModelElement.template |
        UML:ModelElement.templateParameter |
        UML:ModelElement.implementation |
        UML:ModelElement.view |

```

```

UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:Action.request | UML:Action.message |
UML:Action.actionSequence |
UML:CreateAction.instantiation |
UML:ModelElement.taggedValue |
UML:Action.actualArgument)* >

<!ATTLIST UML:CreateAction
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  isAsynchronous (false | true) #IMPLIED
  script CDATA #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED
  partition IDREFS #IMPLIED
  request IDREFS #IMPLIED
  message IDREFS #IMPLIED
  actionSequence IDREFS #IMPLIED
  instantiation IDREFS #IMPLIED
  %XMI.element.att;
  %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.CallAction -->
<!-- _____ -->

<!ELEMENT UML:CallAction.mode EMPTY >
<!ATTLIST UML:CallAction.mode
  xmi.value (synchronous | asynchronous) #REQUIRED
>

<!ELEMENT UML:CallAction (UML:ModelElement.name |
  UML:ModelElement.visibility |

```

```

UML:Action.recurrence | UML:Action.target |
UML:Action.isAsynchronous | UML:Action.script |
UML:CallAction.mode | XMI.extension |
UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:Action.request | UML:Action.message |
UML:Action.actionSequence |
UML:ModelElement.taggedValue |
UML:Action.actualArgument)* >

<!-- UML:CallAction
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isAsynchronous (false | true) #IMPLIED
    script CDATA #IMPLIED
    mode (synchronous | asynchronous) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    request IDREFS #IMPLIED
    message IDREFS #IMPLIED
    actionSequence IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.LocalInvocation -->
<!-- _____ -->

```

```

<!ELEMENT UML:LocalInvocation (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:Action.recurrence | UML:Action.target |
                                UML:Action.isAsynchronous |
                                UML:Action.script | XMI:extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:Action.request | UML:Action.message |
                                UML:Action.actionSequence |
                                UML:ModelElement.taggedValue |
                                UML:Action.actualArgument)* >

<!-- UML:LocalInvocation
      name CDATA #IMPLIED
      visibility (public | protected | private) #IMPLIED
      isAsynchronous (false | true) #IMPLIED
      script CDATA #IMPLIED
      binding IDREFS #IMPLIED
      template IDREFS #IMPLIED
      templateParameter IDREFS #IMPLIED
      implementation IDREFS #IMPLIED
      view IDREFS #IMPLIED
      presentation IDREFS #IMPLIED
      namespace IDREFS #IMPLIED
      constraint IDREFS #IMPLIED
      requirement IDREFS #IMPLIED
      provision IDREFS #IMPLIED
      stereotype IDREFS #IMPLIED
      elementReference IDREFS #IMPLIED
      collaboration IDREFS #IMPLIED
      behavior IDREFS #IMPLIED
      partition IDREFS #IMPLIED
      request IDREFS #IMPLIED
      message IDREFS #IMPLIED
      actionSequence IDREFS #IMPLIED
      %XMI.element.att;
      %XMI.link.att;
-->

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.ReturnAction -->
<!-- _____ -->

```



```

<!ELEMENT UML:ReturnAction (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:Action.recurrence | UML:Action.target |
    UML:Action.isAsynchronous |
    UML:Action.script | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:Action.request | UML:Action.message |
    UML:Action.actionSequence |
    UML:ModelElement.taggedValue |
    UML:Action.actualArgument)* >

<!-- UML:ReturnAction
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isAsynchronous (false | true) #IMPLIED
    script CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    request IDREFS #IMPLIED
    message IDREFS #IMPLIED
    actionSequence IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
-->
<!-- CLASS: Behavioral_Elements.Common_Behavior.SendAction -->

```

```

<!-- _____ -->

<!ELEMENT UML:SendAction (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:Action.recurrence | UML:Action.target |
    UML:Action.isAsynchronous | UML:Action.script |
    XMI.extension | UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:Action.request | UML:Action.message |
    UML:Action.actionSequence |
    UML:ModelElement.taggedValue |
    UML:Action.actualArgument)* >

<!ATTLIST UML:SendAction
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isAsynchronous (false | true) #IMPLIED
    script CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    request IDREFS #IMPLIED
    message IDREFS #IMPLIED
    actionSequence IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;

>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.UninterpretedAction -->

```

```

<!-- _____ -->

<!ELEMENT UML:UninterpretedAction.body (#PCDATA | XMI.reference)* >

<!ELEMENT UML:UninterpretedAction (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:Action.recurrence |
                                UML:Action.target |
                                UML:Action.isAsynchronous |
                                UML:Action.script |
                                UML:UninterpretedAction.body |
                                XMI.extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:Action.request |
                                UML:Action.message |
                                UML:Action.actionSequence |
                                UML:ModelElement.taggedValue |
                                UML:Action.actualArgument)* >

<!ATTLIST UML:UninterpretedAction
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isAsynchronous (false | true) #IMPLIED
    script CDATA #IMPLIED
    body CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    request IDREFS #IMPLIED
    message IDREFS #IMPLIED

```

```

        actionSequence IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.TerminateAction -->
<!-- _____ -->

<!ELEMENT UML:TerminateAction (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:Action.recurrence | UML:Action.target |
                                UML:Action.isAsynchronous |
                                UML:Action.script | XMI:extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:Action.request | UML:Action.message |
                                UML:Action.actionSequence |
                                UML:ModelElement.taggedValue |
                                UML:Action.actualArgument)* >

<!ATTLIST UML:TerminateAction
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isAsynchronous (false | true) #IMPLIED
    script CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    request IDREFS #IMPLIED

```

```

        message IDREFS #IMPLIED
        actionSequence IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.DestroyAction -->
<!-- _____ -->

<!ELEMENT UML:DestroyAction (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:Action.recurrence | UML:Action.target |
                                UML:Action.isAsynchronous |
                                UML:Action.script | XMI:extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:Action.request | UML:Action.message |
                                UML:Action.actionSequence |
                                UML:ModelElement.taggedValue |
                                UML:Action.actualArgument)* >

<!ATTLIST UML:DestroyAction
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isAsynchronous (false | true) #IMPLIED
    script CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED

```

```

        request IDREFS #IMPLIED
        message IDREFS #IMPLIED
        actionSequence IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Link -->
<!-- _____ -->

<!ELEMENT UML:Link.association (UML:Association)* >

<!ELEMENT UML:Link (UML:ModelElement.name | UML:ModelElement.visibility |
                    XMI:extension | UML:ModelElement.binding |
                    UML:ModelElement.template |
                    UML:ModelElement.templateParameter |
                    UML:ModelElement.implementation |
                    UML:ModelElement.view |
                    UML:ModelElement.presentation |
                    UML:ModelElement.namespace |
                    UML:ModelElement.constraint |
                    UML:ModelElement.requirement |
                    UML:ModelElement.provision |
                    UML:ModelElement.stereotype |
                    UML:ModelElement.elementReference |
                    UML:ModelElement.collaboration |
                    UML:ModelElement.behavior |
                    UML:ModelElement.partition | UML:Link.association |
                    UML:ModelElement.taggedValue | UML:Link.linkRole)* >

<!ATTLIST UML:Link
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    association IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->

```

```

<!--                                     -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.LinkEnd             -->
<!-- _____ -->

<!ELEMENT UML:LinkEnd.link (UML:Link)* >

<!ELEMENT UML:LinkEnd.associationEnd (UML:AssociationEnd)* >

<!ELEMENT UML:LinkEnd.instance (UML:Instance)* >

<!ELEMENT UML:LinkEnd (UML:ModelElement.name |
                        UML:ModelElement.visibility | XMI:extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition | UML:LinkEnd.link |
                        UML:LinkEnd.associationEnd | UML:LinkEnd.instance |
                        UML:ModelElement.taggedValue)* >

<!-- ATTLIST UML:LinkEnd
      name CDATA #IMPLIED
      visibility (public | protected | private) #IMPLIED
      binding IDREFS #IMPLIED
      template IDREFS #IMPLIED
      templateParameter IDREFS #IMPLIED
      implementation IDREFS #IMPLIED
      view IDREFS #IMPLIED
      presentation IDREFS #IMPLIED
      namespace IDREFS #IMPLIED
      constraint IDREFS #IMPLIED
      requirement IDREFS #IMPLIED
      provision IDREFS #IMPLIED
      stereotype IDREFS #IMPLIED
      elementReference IDREFS #IMPLIED
      collaboration IDREFS #IMPLIED
      behavior IDREFS #IMPLIED
      partition IDREFS #IMPLIED
      link IDREFS #IMPLIED
      associationEnd IDREFS #IMPLIED
      instance IDREFS #IMPLIED
      %XMI.element.att;
      %XMI.link.att;
-->

<!-- _____ -->

```

```

<!--                                     -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Instance           -->
<!-- _____ -->

<!ELEMENT UML:Instance.linkEnd (UML:LinkEnd)* >

<!ELEMENT UML:Instance.messageInstance3 (UML:MessageInstance)* >

<!ELEMENT UML:Instance.messageInstance2 (UML:MessageInstance)* >

<!ELEMENT UML:Instance.messageInstance (UML:MessageInstance)* >

<!ELEMENT UML:Instance.attributeLink (UML:AttributeLink)* >

<!ELEMENT UML:Instance.classifier (UML:Classifier)* >

<!ELEMENT UML:Instance (UML:ModelElement.name |
                        UML:ModelElement.visibility | XMI:extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:Instance.linkEnd |
                        UML:Instance.messageInstance3 |
                        UML:Instance.messageInstance2 |
                        UML:Instance.messageInstance |
                        UML:Instance.attributeLink |
                        UML:Instance.classifier |
                        UML:ModelElement.taggedValue |
                        UML:Instance.slot)* >

<!-- ATTLIST UML:Instance
      name CDATA #IMPLIED
      visibility (public | protected | private) #IMPLIED
      binding IDREFS #IMPLIED
      template IDREFS #IMPLIED
      templateParameter IDREFS #IMPLIED
      implementation IDREFS #IMPLIED
      view IDREFS #IMPLIED
      presentation IDREFS #IMPLIED
      namespace IDREFS #IMPLIED
      constraint IDREFS #IMPLIED
      requirement IDREFS #IMPLIED
      provision IDREFS #IMPLIED
      stereotype IDREFS #IMPLIED
-->

```



```

        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        linkEnd IDREFS #IMPLIED
        messageInstance3 IDREFS #IMPLIED
        messageInstance2 IDREFS #IMPLIED
        messageInstance IDREFS #IMPLIED
        attributeLink IDREFS #IMPLIED
        classifier IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.AttributeLink -->
<!-- _____ -->

<!ELEMENT UML:AttributeLink.instance (UML:Instance)* >

<!ELEMENT UML:AttributeLink.attribute (UML:Attribute)* >

<!ELEMENT UML:AttributeLink.value (UML:Instance)* >

<!ELEMENT UML:AttributeLink (UML:ModelElement.name |
                                UML:ModelElement.visibility | XMI:extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:AttributeLink.instance |
                                UML:AttributeLink.attribute |
                                UML:AttributeLink.value |
                                UML:ModelElement.taggedValue)* >

<!-- ATTLIST UML:AttributeLink
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
-->

```

```

        namespace IDREFS #IMPLIED
        constraint IDREFS #IMPLIED
        requirement IDREFS #IMPLIED
        provision IDREFS #IMPLIED
        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        instance IDREFS #IMPLIED
        attribute IDREFS #IMPLIED
        value IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.Object -->
<!-- _____ -->

<!ELEMENT UML:Object (UML:ModelElement.name |
    UML:ModelElement.visibility | XMI:extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition | UML:Instance.linkEnd |
    UML:Instance.messageInstance3 |
    UML:Instance.messageInstance2 |
    UML:Instance.messageInstance |
    UML:Instance.attributeLink |
    UML:Instance.classifier |
    UML:ModelElement.taggedValue | UML:Instance.slot)* >

<!ATTLIST UML:Object
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED

```

```

        requirement IDREFS #IMPLIED
        provision IDREFS #IMPLIED
        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        linkEnd IDREFS #IMPLIED
        messageInstance3 IDREFS #IMPLIED
        messageInstance2 IDREFS #IMPLIED
        messageInstance IDREFS #IMPLIED
        attributeLink IDREFS #IMPLIED
        classifier IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.DataValue -->
<!-- _____ -->

<!ELEMENT UML:DataValue (UML:ModelElement.name |
                        UML:ModelElement.visibility | XMI:extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:Instance.linkEnd |
                        UML:Instance.messageInstance3 |
                        UML:Instance.messageInstance2 |
                        UML:Instance.messageInstance |
                        UML:Instance.attributeLink |
                        UML:Instance.classifier |
                        UML:ModelElement.taggedValue |
                        UML:Instance.slot)* >

<!-- ATTLIST UML:DataValue
        name CDATA #IMPLIED
        visibility (public | protected | private) #IMPLIED
        binding IDREFS #IMPLIED
        template IDREFS #IMPLIED
        templateParameter IDREFS #IMPLIED
        implementation IDREFS #IMPLIED
        view IDREFS #IMPLIED

```

```

presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
linkEnd IDREFS #IMPLIED
messageInstance3 IDREFS #IMPLIED
messageInstance2 IDREFS #IMPLIED
messageInstance IDREFS #IMPLIED
attributeLink IDREFS #IMPLIED
classifier IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.LinkObject -->
<!-- _____ -->

<!ELEMENT UML:LinkObject (UML:ModelElement.name |
                        UML:ModelElement.visibility | XMI:extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:Instance.linkEnd |
                        UML:Instance.messageInstance3 |
                        UML:Instance.messageInstance2 |
                        UML:Instance.messageInstance |
                        UML:Instance.attributeLink |
                        UML:Instance.classifier | UML:Link.association |
                        UML:ModelElement.taggedValue |
                        UML:Instance.slot | UML:Link.linkRole)* >

<!ATTLIST UML:LinkObject
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED

```

```

templateParameter IDREFS #IMPLIED
implementation IDREFS #IMPLIED
view IDREFS #IMPLIED
presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
linkEnd IDREFS #IMPLIED
messageInstance3 IDREFS #IMPLIED
messageInstance2 IDREFS #IMPLIED
messageInstance IDREFS #IMPLIED
attributeLink IDREFS #IMPLIED
classifier IDREFS #IMPLIED
association IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Common_Behavior.MessageInstance -->
<!-- _____ -->

<!ELEMENT UML:MessageInstance.specification (UML:Request)* >

<!ELEMENT UML:MessageInstance.sender (UML:Instance)* >

<!ELEMENT UML:MessageInstance.receiver (UML:Instance)* >

<!ELEMENT UML:MessageInstance.argument (UML:Instance)* >

<!ELEMENT UML:MessageInstance (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                XMI:extension | UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:MessageInstance.specification |

```

```

                                UML:MessageInstance.sender |
                                UML:MessageInstance.receiver |
                                UML:MessageInstance.argument |
                                UML:ModelElement.taggedValue)* >

<!-- UML:MessageInstance
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    sender IDREFS #IMPLIED
    receiver IDREFS #IMPLIED
    argument IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Collaborations.Collaboration -->
<!-- _____ -->

<!-- UML:Collaboration.representedClassifier (UML:Classifier)* >

<!-- UML:Collaboration.representedOperation (UML:Operation)* >

<!-- UML:Collaboration.constrainingElement (UML:ModelElement)* >

<!-- UML:Collaboration (UML:ModelElement.name |
                                UML:ModelElement.visibility | XMI:extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |

```

```

UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:Collaboration.representedClassifier |
UML:Collaboration.representedOperation |
UML:Collaboration.constrainingElement |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Collaboration.interaction)* >

<!-- UML:Collaboration
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED
  partition IDREFS #IMPLIED
  representedClassifier IDREFS #IMPLIED
  representedOperation IDREFS #IMPLIED
  constrainingElement IDREFS #IMPLIED
  %XMI.element.att;
  %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Collaborations.Interaction -->
<!-- _____ -->

<!-- UML:Interaction.context (UML:Collaboration)* >

<!-- UML:Interaction (UML:ModelElement.name |
UML:ModelElement.visibility | XMI.extension |
UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |

```

```

        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition |
        UML:Interaction.context |
        UML:ModelElement.taggedValue |
        UML:Interaction.message)* >
<!ATTLIST UML:Interaction
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    context IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Behavioral_Elements.Collaborations.AssociationRole -->
<!-- _____ -->

<!ELEMENT UML:AssociationRole.multiplicity (#PCDATA | XMI.reference)* >

<!ELEMENT UML:AssociationRole.base (UML:Association)* >

<!ELEMENT UML:AssociationRole.namespace (UML:Collaboration)* >

<!ELEMENT UML:AssociationRole (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:GeneralizableElement.isRoot |
    UML:GeneralizableElement.isLeaf |
    UML:GeneralizableElement.isAbstract |
    UML:AssociationRole.multiplicity |
    XMI.extension | UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |

```



```

UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:GeneralizableElement.generalization |
UML:GeneralizableElement.specialization |
UML:Association.link |
UML:Association.associationEnd |
UML:AssociationRole.base |
UML:AssociationRole.namespace |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Association.connection)* >

<!ATTLIST UML:AssociationRole
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  isRoot (false | true) #IMPLIED
  isLeaf (false | true) #IMPLIED
  isAbstract (false | true) #IMPLIED
  multiplicity CDATA #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  ModelElement.namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED
  partition IDREFS #IMPLIED
  generalization IDREFS #IMPLIED
  specialization IDREFS #IMPLIED
  link IDREFS #IMPLIED
  associationEnd IDREFS #IMPLIED
  base IDREFS #IMPLIED
  AssociationRole.namespace IDREFS #IMPLIED
  %XMI.element.att;
  %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Collaborations.AssociationEndRole -->
<!-- _____ -->

<!ELEMENT UML:AssociationEndRole.associationRole
(UML:AssociationRole)* >

```

```

<!ELEMENT UML:AssociationEndRole.base (UML:AssociationEnd)* >

<!ELEMENT UML:AssociationEndRole (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:AssociationEnd.isNavigable |
    UML:AssociationEnd.isOrdered |
    UML:AssociationEnd.aggregation |
    UML:AssociationEnd.multiplicity |
    UML:AssociationEnd.changeable |
    UML:AssociationEnd.targetScope |
    XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:AssociationEnd.type |
    UML:AssociationEnd.specification |
    UML:AssociationEnd.association |
    UML:AssociationEnd.linkEnd |
    UML:AssociationEnd.associationEndRole |
    UML:AssociationEndRole.associationRole |
    UML:AssociationEndRole.base |
    UML:ModelElement.taggedValue |
    UML:AssociationEnd.qualifier)* >

<!ATTLIST UML:AssociationEndRole
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isNavigable (false | true) #IMPLIED
    isOrdered (false | true) #IMPLIED
    aggregation (none | shared | composite) #IMPLIED
    multiplicity CDATA #IMPLIED
    changeable (none | frozen | addOnly) #IMPLIED
    targetScope (classifier | instance) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED

```

```

        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        type IDREFS #IMPLIED
        specification IDREFS #IMPLIED
        association IDREFS #IMPLIED
        linkEnd IDREFS #IMPLIED
        associationEndRole IDREFS #IMPLIED
        associationRole IDREFS #IMPLIED
        base IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Behavioral_Elements.Collaborations.Message -->
<!-- _____ -->

<!ELEMENT UML:Message.interaction (UML:Interaction)* >

<!ELEMENT UML:Message.predecessor (UML:Message)* >

<!ELEMENT UML:Message.message2 (UML:Message)* >

<!ELEMENT UML:Message.message (UML:Message)* >

<!ELEMENT UML:Message.activator (UML:Message)* >

<!ELEMENT UML:Message.action (UML:Action)* >

<!ELEMENT UML:Message.receiver (UML:ClassifierRole)* >

<!ELEMENT UML:Message.sender (UML:ClassifierRole)* >

<!ELEMENT UML:Message (UML:ModelElement.name |
                        UML:ModelElement.visibility | XMI:extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:Message.interaction | UML:Message.predecessor |
                        UML:Message.message2 | UML:Message.message |

```

```

                                UML:Message.activator | UML:Message.action |
                                UML:Message.receiver | UML:Message.sender |
                                UML:ModelElement.taggedValue)* >
<!-- UML:Message
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    interaction IDREFS #IMPLIED
    predecessor IDREFS #IMPLIED
    message2 IDREFS #IMPLIED
    message IDREFS #IMPLIED
    activator IDREFS #IMPLIED
    action IDREFS #IMPLIED
    receiver IDREFS #IMPLIED
    sender IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Collaborations.ClassifierRole -->
<!-- _____ -->

<!-- UML:ClassifierRole.multiplicity (#PCDATA | XMI.reference)* >

<!-- UML:ClassifierRole.associationEndRole
                                (UML:AssociationEndRole)* >

<!-- UML:ClassifierRole.namespace (UML:Collaboration)* >

<!-- UML:ClassifierRole.message2 (UML:Message)* >

<!-- UML:ClassifierRole.message (UML:Message)* >

<!-- UML:ClassifierRole.base (UML:Classifier)* >

<!-- UML:ClassifierRole.availableFeature (UML:Feature)* >

<!-- UML:ClassifierRole (UML:ModelElement.name |
                                UML:ModelElement.visibility |

```

```

UML:GeneralizableElement.isRoot |
UML:GeneralizableElement.isLeaf |
UML:GeneralizableElement.isAbstract |
UML:ClassifierRole.multiplicity |
XMI:extension | UML:ModelElement.binding |
UML:ModelElement.template |
UML:ModelElement.templateParameter |
UML:ModelElement.implementation |
UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:GeneralizableElement.generalization |
UML:GeneralizableElement.specialization |
UML:Classifier.parameter |
UML:Classifier.structuralFeature |
UML:Classifier.specification |
UML:Classifier.realization |
UML:Classifier.associationEnd |
UML:Classifier.participant |
UML:Classifier.createAction |
UML:Classifier.instance |
UML:Classifier.collaboration |
UML:Classifier.classifierRole |
UML:Classifier.classifierInState |
UML:ClassifierRole.associationEndRole |
UML:ClassifierRole.namespace |
UML:ClassifierRole.message2 |
UML:ClassifierRole.message |
UML:ClassifierRole.base |
UML:ClassifierRole.availableFeature |
UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Classifier.feature)* >

<!ATTLIST UML:ClassifierRole
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  isRoot (false | true) #IMPLIED
  isLeaf (false | true) #IMPLIED
  isAbstract (false | true) #IMPLIED
  multiplicity CDATA #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  ModelElement.namespace IDREFS #IMPLIED

```

```

constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
ModelElement.collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
generalization IDREFS #IMPLIED
specialization IDREFS #IMPLIED
parameter IDREFS #IMPLIED
structuralFeature IDREFS #IMPLIED
specification IDREFS #IMPLIED
realization IDREFS #IMPLIED
associationEnd IDREFS #IMPLIED
participant IDREFS #IMPLIED
createAction IDREFS #IMPLIED
instance IDREFS #IMPLIED
Classifier.collaboration IDREFS #IMPLIED
classifierRole IDREFS #IMPLIED
classifierInState IDREFS #IMPLIED
associationEndRole IDREFS #IMPLIED
ClassifierRole.namespace IDREFS #IMPLIED
message2 IDREFS #IMPLIED
message IDREFS #IMPLIED
base IDREFS #IMPLIED
availableFeature IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Use_Cases.Actor -->
<!-- _____ -->

<!ELEMENT UML:Actor (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:GeneralizableElement.isRoot |
    UML:GeneralizableElement.isLeaf |
    UML:GeneralizableElement.isAbstract | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |

```

```

        UML:ModelElement.partition |
        UML:GeneralizableElement.generalization |
        UML:GeneralizableElement.specialization |
        UML:Classifier.parameter |
        UML:Classifier.structuralFeature |
        UML:Classifier.specification |
        UML:Classifier.realization |
        UML:Classifier.associationEnd |
        UML:Classifier.participant |
        UML:Classifier.createAction |
        UML:Classifier.instance |
        UML:Classifier.collaboration |
        UML:Classifier.classifierRole |
        UML:Classifier.classifierInState |
        UML:ModelElement.taggedValue |
        UML:Namespace.ownedElement |
        UML:Classifier.feature)* >
<!ATTLIST UML:Actor
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    ModelElement.collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    parameter IDREFS #IMPLIED
    structuralFeature IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    realization IDREFS #IMPLIED
    associationEnd IDREFS #IMPLIED
    participant IDREFS #IMPLIED
    createAction IDREFS #IMPLIED
    instance IDREFS #IMPLIED
    Classifier.collaboration IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

```

```

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Behavioral_Elements.Use_Cases.UseCase -->
<!-- _____ -->

<!ELEMENT UML:UseCase.extensionPoint (#PCDATA | XMI.reference)* >

<!ELEMENT UML:UseCase (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:GeneralizableElement.isRoot |
    UML:GeneralizableElement.isLeaf |
    UML:GeneralizableElement.isAbstract |
    UML:UseCase.extensionPoint | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition |
    UML:GeneralizableElement.generalization |
    UML:GeneralizableElement.specialization |
    UML:Classifier.parameter |
    UML:Classifier.structuralFeature |
    UML:Classifier.specification |
    UML:Classifier.realization |
    UML:Classifier.associationEnd |
    UML:Classifier.participant |
    UML:Classifier.createAction |
    UML:Classifier.instance |
    UML:Classifier.collaboration |
    UML:Classifier.classifierRole |
    UML:Classifier.classifierInState |
    UML:ModelElement.taggedValue |
    UML:Namespace.ownedElement |
    UML:Classifier.feature)* >

<!ATTLIST UML:UseCase
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    extensionPoint CDATA #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED

```



```

view IDREFS #IMPLIED
presentation IDREFS #IMPLIED
namespace IDREFS #IMPLIED
constraint IDREFS #IMPLIED
requirement IDREFS #IMPLIED
provision IDREFS #IMPLIED
stereotype IDREFS #IMPLIED
elementReference IDREFS #IMPLIED
ModelElement.collaboration IDREFS #IMPLIED
behavior IDREFS #IMPLIED
partition IDREFS #IMPLIED
generalization IDREFS #IMPLIED
specialization IDREFS #IMPLIED
parameter IDREFS #IMPLIED
structuralFeature IDREFS #IMPLIED
specification IDREFS #IMPLIED
realization IDREFS #IMPLIED
associationEnd IDREFS #IMPLIED
participant IDREFS #IMPLIED
createAction IDREFS #IMPLIED
instance IDREFS #IMPLIED
Classifier.collaboration IDREFS #IMPLIED
classifierRole IDREFS #IMPLIED
classifierInState IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.Use_Cases.UseCaseInstance -->
<!-- _____ -->

<!ELEMENT UML:UseCaseInstance (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                XMI:extension | UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:Instance.linkEnd |
                                UML:Instance.messageInstance3 |
                                UML:Instance.messageInstance2 |
                                UML:Instance.messageInstance |
                                UML:Instance.attributeLink |

```

```

UML:Instance.classifier |
UML:ModelElement.taggedValue |
UML:Instance.slot)* >

<!ATTLIST UML:UseCaseInstance
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED
  partition IDREFS #IMPLIED
  linkEnd IDREFS #IMPLIED
  messageInstance3 IDREFS #IMPLIED
  messageInstance2 IDREFS #IMPLIED
  messageInstance IDREFS #IMPLIED
  attributeLink IDREFS #IMPLIED
  classifier IDREFS #IMPLIED
  %XMI.element.att;
  %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.StateMachine -->
<!-- _____ -->

<!ELEMENT UML:StateMachine.context (UML:ModelElement)* >

<!ELEMENT UML:StateMachine.submachineState (UML:SubmachineState)* >

<!ELEMENT UML:StateMachine (UML:ModelElement.name |
  UML:ModelElement.visibility | XMI:extension |
  UML:ModelElement.binding |
  UML:ModelElement.template |
  UML:ModelElement.templateParameter |
  UML:ModelElement.implementation |
  UML:ModelElement.view |
  UML:ModelElement.presentation |
  UML:ModelElement.namespace |
  UML:ModelElement.constraint |
  UML:ModelElement.requirement |
  UML:ModelElement.provision |
  UML:ModelElement.stereotype |
  UML:ModelElement.elementReference |
  UML:ModelElement.collaboration |

```

```

        UML:ModelElement.behavior |
        UML:ModelElement.partition |
        UML:StateMachine.context |
        UML:StateMachine.submachineState |
        UML:ModelElement.taggedValue |
        UML:StateMachine.top |
        UML:StateMachine.transitions)* >
<!-- UML:StateMachine
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    context IDREFS #IMPLIED
    submachineState IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.Guard -->
<!-- _____ -->

<!-- UML:Guard.expression (UML:BooleanExpression) >

<!-- UML:Guard.transition (UML:Transition)* >

<!-- UML:Guard (UML:ModelElement.name |
        UML:ModelElement.visibility | UML:Guard.expression |
        XMI.extension | UML:ModelElement.binding |
        UML:ModelElement.template |
        UML:ModelElement.templateParameter |
        UML:ModelElement.implementation |
        UML:ModelElement.view |
        UML:ModelElement.presentation |
        UML:ModelElement.namespace |
        UML:ModelElement.constraint |
        UML:ModelElement.requirement |
        UML:ModelElement.provision |
        UML:ModelElement.stereotype |
        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |

```

```

                                UML:ModelElement.behavior |
                                UML:ModelElement.partition | UML:Guard.transition |
                                UML:ModelElement.taggedValue)* >
<!-- UML:Guard
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    transition IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.StateVertex -->
<!-- _____ -->

<!-- UML:StateVertex.outgoing (UML:Transition)* >

<!-- UML:StateVertex.incoming (UML:Transition)* >

<!-- UML:StateVertex.parent (UML:CompositeState)* >

<!-- UML:StateVertex (UML:ModelElement.name |
                                UML:ModelElement.visibility | XMI.extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:StateVertex.outgoing |

```

```

                                UML:StateVertex.incoming |
                                UML:StateVertex.parent |
                                UML:ModelElement.taggedValue)* >
<!-- UML:StateVertex
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    outgoing IDREFS #IMPLIED
    incoming IDREFS #IMPLIED
    parent IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.Transition -->
<!-- _____ -->

<!-- UML:Transition.source (UML:StateVertex)* >

<!-- UML:Transition.target (UML:StateVertex)* >

<!-- UML:Transition.statemachine (UML:StateMachine)* >

<!-- UML:Transition.state (UML:State)* >

<!-- UML:Transition.trigger (UML:Event)* >

<!-- UML:Transition (UML:ModelElement.name |
                                UML:ModelElement.visibility | XMI.extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |

```

```

                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:Transition.source | UML:Transition.target |
                                UML:Transition.statemachine |
                                UML:Transition.state | UML:Transition.trigger |
                                UML:ModelElement.taggedValue |
                                UML:Transition.guard | UML:Transition.effect)* >
<!-- UML:Transition
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    source IDREFS #IMPLIED
    target IDREFS #IMPLIED
    statemachine IDREFS #IMPLIED
    state IDREFS #IMPLIED
    trigger IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.PseudoState -->
<!-- _____ -->

<!-- UML:PseudoState.kind EMPTY >
<!-- UML:PseudoState.kind
    xmi.value (initial | deepHistory | shallowHistory | join | fork
| branch | final) #REQUIRED
>

<!-- UML:PseudoState (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:PseudoState.kind | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |

```

```

        UML:ModelElement.view |
        UML:ModelElement.presentation |
        UML:ModelElement.namespace |
        UML:ModelElement.constraint |
        UML:ModelElement.requirement |
        UML:ModelElement.provision |
        UML:ModelElement.stereotype |
        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition |
        UML:StateVertex.outgoing |
        UML:StateVertex.incoming |
        UML:StateVertex.parent |
        UML:ModelElement.taggedValue)* >
<!ATTLIST UML:PseudoState
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    kind (initial | deepHistory | shallowHistory | join | fork | branch
| final) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    outgoing IDREFS #IMPLIED
    incoming IDREFS #IMPLIED
    parent IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.State -->
<!-- _____ -->

<!ELEMENT UML:State.stateMachine (UML:StateMachine)* >

<!ELEMENT UML:State.deferredEvent (UML:Event)* >

<!ELEMENT UML:State.classifierInState (UML:ClassifierInState)* >

<!ELEMENT UML:State (UML:ModelElement.name |
                    UML:ModelElement.visibility | XMI:extension |

```

```

        UML:ModelElement.binding |
        UML:ModelElement.template |
        UML:ModelElement.templateParameter |
        UML:ModelElement.implementation |
        UML:ModelElement.view |
        UML:ModelElement.presentation |
        UML:ModelElement.namespace |
        UML:ModelElement.constraint |
        UML:ModelElement.requirement |
        UML:ModelElement.provision |
        UML:ModelElement.stereotype |
        UML:ModelElement.elementReference |
        UML:ModelElement.collaboration |
        UML:ModelElement.behavior |
        UML:ModelElement.partition |
        UML:StateVertex.outgoing | UML:StateVertex.incoming |
        UML:StateVertex.parent | UML:State.stateMachine |
        UML:State.deferredEvent |
        UML:State.classifierInState |
        UML:ModelElement.taggedValue |
        UML:State.internalTransition | UML:State.entry |
        UML:State.exit)* >
<!-- UML:State
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    outgoing IDREFS #IMPLIED
    incoming IDREFS #IMPLIED
    parent IDREFS #IMPLIED
    stateMachine IDREFS #IMPLIED
    deferredEvent IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.CompositeState -->
<!-- _____ -->

```



```

<!ELEMENT UML:CompositeState.isConcurrent EMPTY >
<!ATTLIST UML:CompositeState.isConcurrent
          xmi.value (false | true) #REQUIRED
>

<!ELEMENT UML:CompositeState (UML:ModelElement.name |
                               UML:ModelElement.visibility |
                               UML:CompositeState.isConcurrent |
                               XMI.extension | UML:ModelElement.binding |
                               UML:ModelElement.template |
                               UML:ModelElement.templateParameter |
                               UML:ModelElement.implementation |
                               UML:ModelElement.view |
                               UML:ModelElement.presentation |
                               UML:ModelElement.namespace |
                               UML:ModelElement.constraint |
                               UML:ModelElement.requirement |
                               UML:ModelElement.provision |
                               UML:ModelElement.stereotype |
                               UML:ModelElement.elementReference |
                               UML:ModelElement.collaboration |
                               UML:ModelElement.behavior |
                               UML:ModelElement.partition |
                               UML:StateVertex.outgoing |
                               UML:StateVertex.incoming |
                               UML:StateVertex.parent |
                               UML:State.stateMachine |
                               UML:State.deferredEvent |
                               UML:State.classifierInState |
                               UML:ModelElement.taggedValue |
                               UML:State.internalTransition |
                               UML:State.entry | UML:State.exit |
                               UML:CompositeState.substate)* >

<!ATTLIST UML:CompositeState
          name CDATA #IMPLIED
          visibility (public | protected | private) #IMPLIED
          isConcurrent (false | true) #IMPLIED
          binding IDREFS #IMPLIED
          template IDREFS #IMPLIED
          templateParameter IDREFS #IMPLIED
          implementation IDREFS #IMPLIED
          view IDREFS #IMPLIED
          presentation IDREFS #IMPLIED
          namespace IDREFS #IMPLIED
          constraint IDREFS #IMPLIED
          requirement IDREFS #IMPLIED
          provision IDREFS #IMPLIED
          stereotype IDREFS #IMPLIED
          elementReference IDREFS #IMPLIED
          collaboration IDREFS #IMPLIED
          behavior IDREFS #IMPLIED
          partition IDREFS #IMPLIED
          outgoing IDREFS #IMPLIED
          incoming IDREFS #IMPLIED
          parent IDREFS #IMPLIED

```

```

stateMachine IDREFS #IMPLIED
deferredEvent IDREFS #IMPLIED
classifierInState IDREFS #IMPLIED
%XMI.element.att;
%XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.SimpleState -->
<!-- _____ -->

<!ELEMENT UML:SimpleState (UML:ModelElement.name |
                           UML:ModelElement.visibility | XMI.extension |
                           UML:ModelElement.binding |
                           UML:ModelElement.template |
                           UML:ModelElement.templateParameter |
                           UML:ModelElement.implementation |
                           UML:ModelElement.view |
                           UML:ModelElement.presentation |
                           UML:ModelElement.namespace |
                           UML:ModelElement.constraint |
                           UML:ModelElement.requirement |
                           UML:ModelElement.provision |
                           UML:ModelElement.stereotype |
                           UML:ModelElement.elementReference |
                           UML:ModelElement.collaboration |
                           UML:ModelElement.behavior |
                           UML:ModelElement.partition |
                           UML:StateVertex.outgoing |
                           UML:StateVertex.incoming |
                           UML:StateVertex.parent |
                           UML:State.stateMachine |
                           UML:State.deferredEvent |
                           UML:State.classifierInState |
                           UML:ModelElement.taggedValue |
                           UML:State.internalTransition |
                           UML:State.entry | UML:State.exit)* >

<!ATTLIST UML:SimpleState
  name CDATA #IMPLIED
  visibility (public | protected | private) #IMPLIED
  binding IDREFS #IMPLIED
  template IDREFS #IMPLIED
  templateParameter IDREFS #IMPLIED
  implementation IDREFS #IMPLIED
  view IDREFS #IMPLIED
  presentation IDREFS #IMPLIED
  namespace IDREFS #IMPLIED
  constraint IDREFS #IMPLIED
  requirement IDREFS #IMPLIED
  provision IDREFS #IMPLIED
  stereotype IDREFS #IMPLIED
  elementReference IDREFS #IMPLIED
  collaboration IDREFS #IMPLIED
  behavior IDREFS #IMPLIED

```

```

        partition IDREFS #IMPLIED
        outgoing IDREFS #IMPLIED
        incoming IDREFS #IMPLIED
        parent IDREFS #IMPLIED
        stateMachine IDREFS #IMPLIED
        deferredEvent IDREFS #IMPLIED
        classifierInState IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- -->
<!-- CLASS: Behavioral_Elements.State_Machines.SubmachineState -->
<!-- _____ -->

<!ELEMENT UML:SubmachineState.stateMachine (UML:StateMachine)* >

<!ELEMENT UML:SubmachineState (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                XMI.extension | UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:StateVertex.outgoing |
                                UML:StateVertex.incoming |
                                UML:StateVertex.parent |
                                UML:State.stateMachine |
                                UML:State.deferredEvent |
                                UML:State.classifierInState |
                                UML:SubmachineState.stateMachine |
                                UML:ModelElement.taggedValue |
                                UML:State.internalTransition |
                                UML:State.entry | UML:State.exit)* >

<!ATTLIST UML:SubmachineState
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED

```

```

        constraint IDREFS #IMPLIED
        requirement IDREFS #IMPLIED
        provision IDREFS #IMPLIED
        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        outgoing IDREFS #IMPLIED
        incoming IDREFS #IMPLIED
        parent IDREFS #IMPLIED
        State.stateMachine IDREFS #IMPLIED
        deferredEvent IDREFS #IMPLIED
        classifierInState IDREFS #IMPLIED
        SubmachineState.stateMachine IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.Event -->
<!-- _____ -->

<!ELEMENT UML:Event.state (UML:State)* >

<!ELEMENT UML:Event.transition (UML:Transition)* >

<!ELEMENT UML:Event (UML:ModelElement.name |
                    UML:ModelElement.visibility | XMI:extension |
                    UML:ModelElement.binding |
                    UML:ModelElement.template |
                    UML:ModelElement.templateParameter |
                    UML:ModelElement.implementation |
                    UML:ModelElement.view |
                    UML:ModelElement.presentation |
                    UML:ModelElement.namespace |
                    UML:ModelElement.constraint |
                    UML:ModelElement.requirement |
                    UML:ModelElement.provision |
                    UML:ModelElement.stereotype |
                    UML:ModelElement.elementReference |
                    UML:ModelElement.collaboration |
                    UML:ModelElement.behavior |
                    UML:ModelElement.partition | UML:Event.state |
                    UML:Event.transition |
                    UML:ModelElement.taggedValue)* >

<!ATTLIST UML:Event
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED

```

```

        presentation IDREFS #IMPLIED
        namespace IDREFS #IMPLIED
        constraint IDREFS #IMPLIED
        requirement IDREFS #IMPLIED
        provision IDREFS #IMPLIED
        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        state IDREFS #IMPLIED
        transition IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.SignalEvent -->
<!-- _____ -->

<!ELEMENT UML:SignalEvent.signal (UML:Signal)* >

<!ELEMENT UML:SignalEvent (UML:ModelElement.name |
    UML:ModelElement.visibility | XMI:extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition | UML:Event.state |
    UML:Event.transition | UML:SignalEvent.signal |
    UML:ModelElement.taggedValue)* >

<!ATTLIST UML:SignalEvent
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED

```

```

        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED
        partition IDREFS #IMPLIED
        state IDREFS #IMPLIED
        transition IDREFS #IMPLIED
        signal IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.CallEvent -->
<!-- _____ -->

<!ELEMENT UML:CallEvent.operation (UML:Operation)* >

<!ELEMENT UML:CallEvent (UML:ModelElement.name |
                        UML:ModelElement.visibility | XMI:extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition | UML:Event.state |
                        UML:Event.transition | UML:CallEvent.operation |
                        UML:ModelElement.taggedValue)* >

<!ATTLIST UML:CallEvent
        name CDATA #IMPLIED
        visibility (public | protected | private) #IMPLIED
        binding IDREFS #IMPLIED
        template IDREFS #IMPLIED
        templateParameter IDREFS #IMPLIED
        implementation IDREFS #IMPLIED
        view IDREFS #IMPLIED
        presentation IDREFS #IMPLIED
        namespace IDREFS #IMPLIED
        constraint IDREFS #IMPLIED
        requirement IDREFS #IMPLIED
        provision IDREFS #IMPLIED
        stereotype IDREFS #IMPLIED
        elementReference IDREFS #IMPLIED
        collaboration IDREFS #IMPLIED
        behavior IDREFS #IMPLIED

```

```

        partition IDREFS #IMPLIED
        state IDREFS #IMPLIED
        transition IDREFS #IMPLIED
        operation IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.TimeEvent -->
<!-- _____ -->

<!ELEMENT UML:TimeEvent.duration (UML:TimeExpression) >

<!ELEMENT UML:TimeEvent (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:TimeEvent.duration | XMI.extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition | UML:Event.state |
    UML:Event.transition |
    UML:ModelElement.taggedValue)* >

<!ATTLIST UML:TimeEvent
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    state IDREFS #IMPLIED
    transition IDREFS #IMPLIED

```

```

        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.ChangeEvent -->
<!-- _____ -->

<!ELEMENT UML:ChangeEvent.changeExpression (UML:BooleanExpression) >

<!ELEMENT UML:ChangeEvent (UML:ModelElement.name |
    UML:ModelElement.visibility |
    UML:ChangeEvent.changeExpression |
    XMI.extension | UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |
    UML:ModelElement.view |
    UML:ModelElement.presentation |
    UML:ModelElement.namespace |
    UML:ModelElement.constraint |
    UML:ModelElement.requirement |
    UML:ModelElement.provision |
    UML:ModelElement.stereotype |
    UML:ModelElement.elementReference |
    UML:ModelElement.collaboration |
    UML:ModelElement.behavior |
    UML:ModelElement.partition | UML:Event.state |
    UML:Event.transition |
    UML:ModelElement.taggedValue)* >

<!ATTLIST UML:ChangeEvent
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    state IDREFS #IMPLIED
    transition IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

```



```

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.ActivityModel -->
<!-- _____ -->

<!ELEMENT UML:ActivityModel (UML:ModelElement.name |
                                UML:ModelElement.visibility | XMI.extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:StateMachine.context |
                                UML:StateMachine.submachineState |
                                UML:ModelElement.taggedValue |
                                UML:StateMachine.top |
                                UML:StateMachine.transitions |
                                UML:ActivityModel.partition)* >

<!ATTLIST UML:ActivityModel
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    context IDREFS #IMPLIED
    submachineState IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.Partition -->

```

```

<!-- _____ -->

<!ELEMENT UML:Partition.contents (UML:ModelElement)* >

<!ELEMENT UML:Partition.activityModel (UML:ActivityModel)* >

<!ELEMENT UML:Partition (UML:ModelElement.name |
                        UML:ModelElement.visibility | XMI.extension |
                        UML:ModelElement.binding |
                        UML:ModelElement.template |
                        UML:ModelElement.templateParameter |
                        UML:ModelElement.implementation |
                        UML:ModelElement.view |
                        UML:ModelElement.presentation |
                        UML:ModelElement.namespace |
                        UML:ModelElement.constraint |
                        UML:ModelElement.requirement |
                        UML:ModelElement.provision |
                        UML:ModelElement.stereotype |
                        UML:ModelElement.elementReference |
                        UML:ModelElement.collaboration |
                        UML:ModelElement.behavior |
                        UML:ModelElement.partition |
                        UML:Partition.contents |
                        UML:Partition.activityModel |
                        UML:ModelElement.taggedValue)* >

<!ATTLIST UML:Partition
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    contents IDREFS #IMPLIED
    activityModel IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;

>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.ActionState -->
<!-- _____ -->

```

```

<!ELEMENT UML:ActionState (UML:ModelElement.name |
                           UML:ModelElement.visibility | XMI.extension |
                           UML:ModelElement.binding |
                           UML:ModelElement.template |
                           UML:ModelElement.templateParameter |
                           UML:ModelElement.implementation |
                           UML:ModelElement.view |
                           UML:ModelElement.presentation |
                           UML:ModelElement.namespace |
                           UML:ModelElement.constraint |
                           UML:ModelElement.requirement |
                           UML:ModelElement.provision |
                           UML:ModelElement.stereotype |
                           UML:ModelElement.elementReference |
                           UML:ModelElement.collaboration |
                           UML:ModelElement.behavior |
                           UML:ModelElement.partition |
                           UML:StateVertex.outgoing |
                           UML:StateVertex.incoming |
                           UML:StateVertex.parent |
                           UML:State.stateMachine |
                           UML:State.deferredEvent |
                           UML:State.classifierInState |
                           UML:ModelElement.taggedValue |
                           UML:State.internalTransition |
                           UML:State.entry | UML:State.exit)* >

<!-- UML:ActionState
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    outgoing IDREFS #IMPLIED
    incoming IDREFS #IMPLIED
    parent IDREFS #IMPLIED
    stateMachine IDREFS #IMPLIED
    deferredEvent IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
-->

```

```

<!--                                     -->
<!-- CLASS: Behavioral_Elements.State_Machines.ObjectFlowState      -->
<!-- _____ -->

<!ELEMENT UML:ObjectFlowState.typeState (UML:ClassifierInState)* >

<!ELEMENT UML:ObjectFlowState (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                XMI.extension | UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:StateVertex.outgoing |
                                UML:StateVertex.incoming |
                                UML:StateVertex.parent |
                                UML:State.stateMachine |
                                UML:State.deferredEvent |
                                UML:State.classifierInState |
                                UML:ObjectFlowState.typeState |
                                UML:ModelElement.taggedValue |
                                UML:State.internalTransition |
                                UML:State.entry | UML:State.exit)* >

<!ATTLIST UML:ObjectFlowState
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    outgoing IDREFS #IMPLIED
    incoming IDREFS #IMPLIED
    parent IDREFS #IMPLIED
    stateMachine IDREFS #IMPLIED

```

```

        deferredEvent IDREFS #IMPLIED
        classifierInState IDREFS #IMPLIED
        typeState IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.ClassifierInState -->
<!-- _____ -->

<!ELEMENT UML:ClassifierInState.objectFlowState (UML:ObjectFlowState)* >

<!ELEMENT UML:ClassifierInState.inState (UML:State)* >

<!ELEMENT UML:ClassifierInState.type (UML:Classifier)* >

<!ELEMENT UML:ClassifierInState (UML:ModelElement.name |
                                UML:ModelElement.visibility |
                                UML:GeneralizableElement.isRoot |
                                UML:GeneralizableElement.isLeaf |
                                UML:GeneralizableElement.isAbstract |
                                XMI.extension |
                                UML:ModelElement.binding |
                                UML:ModelElement.template |
                                UML:ModelElement.templateParameter |
                                UML:ModelElement.implementation |
                                UML:ModelElement.view |
                                UML:ModelElement.presentation |
                                UML:ModelElement.namespace |
                                UML:ModelElement.constraint |
                                UML:ModelElement.requirement |
                                UML:ModelElement.provision |
                                UML:ModelElement.stereotype |
                                UML:ModelElement.elementReference |
                                UML:ModelElement.collaboration |
                                UML:ModelElement.behavior |
                                UML:ModelElement.partition |
                                UML:GeneralizableElement.generalization |
                                UML:GeneralizableElement.specialization |
                                UML:Classifier.parameter |
                                UML:Classifier.structuralFeature |
                                UML:Classifier.specification |
                                UML:Classifier.realization |
                                UML:Classifier.associationEnd |
                                UML:Classifier.participant |
                                UML:Classifier.createAction |
                                UML:Classifier.instance |
                                UML:Classifier.collaboration |
                                UML:Classifier.classifierRole |
                                UML:Classifier.classifierInState |
                                UML:ClassifierInState.objectFlowState |
                                UML:ClassifierInState.inState |
                                UML:ClassifierInState.type |

```

```

UML:ModelElement.taggedValue |
UML:Namespace.ownedElement |
UML:Classifier.feature)* >

<!ATTLIST UML:ClassifierInState
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    isRoot (false | true) #IMPLIED
    isLeaf (false | true) #IMPLIED
    isAbstract (false | true) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    ModelElement.collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    generalization IDREFS #IMPLIED
    specialization IDREFS #IMPLIED
    parameter IDREFS #IMPLIED
    structuralFeature IDREFS #IMPLIED
    specification IDREFS #IMPLIED
    realization IDREFS #IMPLIED
    associationEnd IDREFS #IMPLIED
    participant IDREFS #IMPLIED
    createAction IDREFS #IMPLIED
    instance IDREFS #IMPLIED
    Classifier.collaboration IDREFS #IMPLIED
    classifierRole IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    objectFlowState IDREFS #IMPLIED
    inState IDREFS #IMPLIED
    type IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Behavioral_Elements.State_Machines.ActivityState -->
<!-- _____ -->

<!ELEMENT UML:ActivityState (UML:ModelElement.name |
    UML:ModelElement.visibility | XMI:extension |
    UML:ModelElement.binding |
    UML:ModelElement.template |
    UML:ModelElement.templateParameter |
    UML:ModelElement.implementation |

```

```

UML:ModelElement.view |
UML:ModelElement.presentation |
UML:ModelElement.namespace |
UML:ModelElement.constraint |
UML:ModelElement.requirement |
UML:ModelElement.provision |
UML:ModelElement.stereotype |
UML:ModelElement.elementReference |
UML:ModelElement.collaboration |
UML:ModelElement.behavior |
UML:ModelElement.partition |
UML:StateVertex.outgoing |
UML:StateVertex.incoming |
UML:StateVertex.parent |
UML:State.stateMachine |
UML:State.deferredEvent |
UML:State.classifierInState |
UML:SubmachineState.stateMachine |
UML:ModelElement.taggedValue |
UML:State.internalTransition |
UML:State.entry | UML:State.exit)* >

<!ATTLIST UML:ActivityState
    name CDATA #IMPLIED
    visibility (public | protected | private) #IMPLIED
    binding IDREFS #IMPLIED
    template IDREFS #IMPLIED
    templateParameter IDREFS #IMPLIED
    implementation IDREFS #IMPLIED
    view IDREFS #IMPLIED
    presentation IDREFS #IMPLIED
    namespace IDREFS #IMPLIED
    constraint IDREFS #IMPLIED
    requirement IDREFS #IMPLIED
    provision IDREFS #IMPLIED
    stereotype IDREFS #IMPLIED
    elementReference IDREFS #IMPLIED
    collaboration IDREFS #IMPLIED
    behavior IDREFS #IMPLIED
    partition IDREFS #IMPLIED
    outgoing IDREFS #IMPLIED
    incoming IDREFS #IMPLIED
    parent IDREFS #IMPLIED
    State.stateMachine IDREFS #IMPLIED
    deferredEvent IDREFS #IMPLIED
    classifierInState IDREFS #IMPLIED
    SubmachineState.stateMachine IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!ELEMENT UML:Foundation (UML:Auxiliary_Elements | UML:Core |
                           UML:Extension_Mechanisms | UML:Data_Types)* >

<!ATTLIST UML:Foundation
    %XMI.element.att;
    %XMI.link.att;

```

```

>

<!ELEMENT UML:Auxiliary_Elements (UML:Refinement | UML:Usage |
                                   UML:Trace | UML:Binding | UML:Node |
                                   UML:Component | UML:Comment |
                                   UML:ViewElement | UML:Presentation)* >

<!ATTLIST UML:Auxiliary_Elements
           %XMI.element.att;
           %XMI.link.att;
>

<!ELEMENT UML:Core (UML:Element | UML:ModelElement | UML:Namespace |
                    UML:GeneralizableElement | UML:Classifier |
                    UML:Interface | UML:Class | UML:DataType |
                    UML:Feature | UML:StructuralFeature |
                    UML:BehavioralFeature | UML:Operation | UML:Method |
                    UML:Parameter | UML:Constraint | UML:Dependency |
                    UML:Generalization | UML:AssociationEnd |
                    UML:Association | UML:AssociationClass |
                    UML:Attribute)* >

<!ATTLIST UML:Core
           %XMI.element.att;
           %XMI.link.att;
>

<!ELEMENT UML:Extension_Mechanisms (UML:TaggedValue | UML:Stereotype)* >
<!ATTLIST UML:Extension_Mechanisms
           %XMI.element.att;
           %XMI.link.att;
>

<!ELEMENT UML:Data_Types (UML:Enumeration | UML:EnumerationLiteral |
                           UML:Primitive | UML:Structure |
                           UML:MultiplicityRange | UML:Geometry |
                           UML:GraphicMarker | UML:Mapping |
                           UML:Expression | UML:ProcedureExpression |
                           UML:ObjectSetExpression | UML:TimeExpression |
                           UML:BooleanExpression)* >

<!ATTLIST UML:Data_Types
           %XMI.element.att;
           %XMI.link.att;
>

<!ELEMENT UML:Model_Management (UML:Package | UML:Subsystem | UML:Model |
                                 UML:ElementReference)* >

<!ATTLIST UML:Model_Management
           %XMI.element.att;
           %XMI.link.att;
>

<!ELEMENT UML:Behavioral_Elements (UML:Common_Behavior |
                                    UML:Collaborations | UML:Use_Cases |
                                    UML:State_Machines)* >

<!ATTLIST UML:Behavioral_Elements
           %XMI.element.att;

```



```

        %XMI.link.att;
    >

<!ELEMENT UML:Common_Behavior (UML:Request | UML:Signal | UML:Exception |
                                UML:Reception | UML:Argument |
                                UML:ActionSequence | UML:Action |
                                UML:CreateAction | UML:CallAction |
                                UML:LocalInvocation | UML:ReturnAction |
                                UML:SendAction | UML:UninterpretedAction |
                                UML:TerminateAction | UML:DestroyAction |
                                UML:Link | UML:LinkEnd | UML:Instance |
                                UML:AttributeLink | UML:Object |
                                UML:DataValue | UML:LinkObject |
                                UML:MessageInstance)* >

<!ATTLIST UML:Common_Behavior
          %XMI.element.att;
          %XMI.link.att;
>

<!ELEMENT UML:Collaborations (UML:Collaboration | UML:Interaction |
                               UML:AssociationRole |
                               UML:AssociationEndRole | UML:Message |
                               UML:ClassifierRole)* >

<!ATTLIST UML:Collaborations
          %XMI.element.att;
          %XMI.link.att;
>

<!ELEMENT UML:Use_Cases (UML:Actor | UML:UseCase |
                          UML:UseCaseInstance)* >

<!ATTLIST UML:Use_Cases
          %XMI.element.att;
          %XMI.link.att;
>

<!ELEMENT UML:State_Machines (UML:StateMachine | UML:Guard |
                               UML:StateVertex | UML:Transition |
                               UML:PseudoState | UML:State |
                               UML:CompositeState | UML:SimpleState |
                               UML:SubmachineState | UML:Event |
                               UML:SignalEvent | UML:CallEvent |
                               UML:TimeEvent | UML:ChangeEvent |
                               UML:ActivityModel | UML:Partition |
                               UML:ActionState | UML:ObjectFlowState |
                               UML:ClassifierInState |
                               UML:ActivityState)* >

<!ATTLIST UML:State_Machines
          %XMI.element.att;
          %XMI.link.att;
>

```


B.1 Introduction

The MOF 1.1 XMI 1.1 DTD included here uses Rule Set 1 rather than Rule Set 3, which was used for the previous DTD. To use the Corba XML elements that were included in the previous DTD, you will need to add them to the fixed DTD part of this DTD.

Appendix B contains a DTD generated by programmatically applying Rule Set 3 of the DTD Production Rules to a MOF model of itself. This model 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 transferring metamodels, the MOF model itself is treated as a metamodel and 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"?>

<!-- Generated by: XMI Framework 1.0 -->
<!-- Date: Mon Oct 25 17:35:30 PDT 1999 -->

<!-- _____ -->
```

```

<!--                                     -->
<!-- This section of the DTD contains XML declarations required by -->
<!-- XMI.                                     -->
<!-- _____ -->

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

<!ELEMENT XMI (XMI.header?, XMI.content?, XMI.difference*,
               XMI.extensions*) >
<!ATTLIST XMI
          xmi.version CDATA #FIXED "1.1"
          timestamp CDATA #IMPLIED
          verified (true | false) #IMPLIED
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.header contains documentation and identifies the model, -->
<!-- metamodel, and metamodel -->
<!-- _____ -->

<!ELEMENT XMI.header (XMI.documentation?, XMI.model*, XMI.metamodel*,
                     XMI.metamodel*, XMI.import*) >

<!-- _____ -->
<!--                                     -->
<!-- documentation for transfer data -->
<!-- _____ -->

<!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 >

<!-- _____ -->
<!--                                     -->
<!-- XMI.element.att defines the attributes that each XML element -->
<!-- that corresponds to a metamodel class must have to conform to -->
<!-- the XMI specification. -->
<!-- _____ -->

<!ENTITY % XMI.element.att

```

```

        'xmi.id ID #IMPLIED xmi.label CDATA #IMPLIED xmi.uuid
        CDATA #IMPLIED ' >

<!-- _____ -->
<!-- -->
<!-- XMI.link.att defines the attributes that each XML element that -->
<!-- corresponds to a metamodel class must have to enable it to -->
<!-- function as a simple XLink as well as refer to model -->
<!-- constructs within the same XMI file. -->
<!-- _____ -->

<!ENTITY % XMI.link.att
        'href CDATA #IMPLIED xmi.idref IDREF #IMPLIED xml:link
        CDATA #IMPLIED xlink:inline (true | false) #IMPLIED
        xlink:actuate (show | user) #IMPLIED xlink:content-role
        CDATA #IMPLIED xlink:title CDATA #IMPLIED xlink:show
        (embed | replace | new) #IMPLIED xlink:behavior CDATA
        #IMPLIED' >

<!-- _____ -->
<!-- -->
<!-- XMI.model identifies the model(s) being transferred -->
<!-- _____ -->

<!ELEMENT XMI.model ANY >
<!ATTLIST XMI.model
        %XMI.link.att;
        xmi.name CDATA #REQUIRED
        xmi.version CDATA #IMPLIED
>

<!-- _____ -->
<!-- -->
<!-- XMI.metamodel identifies the metamodel(s) for the transferred -->
<!-- data -->
<!-- _____ -->

<!ELEMENT XMI.metamodel ANY >
<!ATTLIST XMI.metamodel
        %XMI.link.att;
        xmi.name CDATA #REQUIRED
        xmi.version CDATA #IMPLIED
>

<!-- _____ -->
<!-- -->
<!-- XMI.metametamodel identifies the metametamodel(s) for the -->
<!-- transferred data -->
<!-- _____ -->

<!ELEMENT XMI.metametamodel ANY >
<!ATTLIST XMI.metametamodel
        %XMI.link.att;
        xmi.name CDATA #REQUIRED

```

```

                xmi.version    CDATA #IMPLIED
>

<!-- _____ -->
<!-- _____ -->
<!-- XMI.metametamodel identifies other files associated with the -->
<!-- transferred data -->
<!-- _____ -->

<!ELEMENT XMI.import ANY >
<!-- ATTLIST XMI.import
                %XMI.link.att;
                xmi.name      CDATA #REQUIRED
                xmi.version    CDATA #IMPLIED
>

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

<!ELEMENT XMI.content ANY >

<!-- _____ -->
<!-- _____ -->
<!-- XMI.extensions contains data to transfer that does not conform -->
<!-- to the metamodel(s) in the header -->
<!-- _____ -->

<!ELEMENT XMI.extensions ANY >
<!-- ATTLIST XMI.extensions
                xmi.extender CDATA #REQUIRED
>

<!-- _____ -->
<!-- _____ -->
<!-- extension contains information related to a specific model -->
<!-- construct that is not defined in the metamodel(s) in the header -->
<!-- _____ -->

<!ELEMENT XMI.extension ANY >
<!-- ATTLIST XMI.extension
                %XMI.element.att;
                %XMI.link.att;
                xmi.extender CDATA #REQUIRED
                xmi.extenderID CDATA #IMPLIED
>

<!-- _____ -->
<!-- _____ -->
<!-- XMI.difference holds XML elements representing differences to a -->
<!-- base model -->
<!-- _____ -->

<!ELEMENT XMI.difference (XMI.difference | XMI.delete | XMI.add |
```

```

                                XMI.replace)* >
<!ATTLIST XMI.difference
            %XMI.element.att;
            %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.delete represents a deletion from a base model -->
<!-- _____ -->

<!ELEMENT XMI.delete EMPTY >
<!ATTLIST XMI.delete
            %XMI.element.att;
            %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.add represents an addition to a base model -->
<!-- _____ -->

<!ELEMENT XMI.add ANY >
<!ATTLIST XMI.add
            %XMI.element.att;
            %XMI.link.att;
            xmi.position CDATA "-1"
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.replace represents the replacement of a model construct -->
<!-- with another model construct in a base model -->
<!-- _____ -->

<!ELEMENT XMI.replace ANY >
<!ATTLIST XMI.replace
            %XMI.element.att;
            %XMI.link.att;
            xmi.position CDATA "-1"
>

<!-- _____ -->
<!--                                     -->
<!-- XMI.reference may be used to refer to data types not defined in -->
<!-- the metamodel -->
<!-- _____ -->

<!ELEMENT XMI.reference ANY >
<!ATTLIST XMI.reference
            %XMI.link.att;
>

<!ATTLIST XMI
            xmlns:MOF CDATA #IMPLIED

```

```

>

<!ELEMENT XMI.any ANY >
<!ATTLIST XMI.any
    %XMI.link.att;
    xmi.type CDATA #IMPLIED
    xmi.name CDATA #IMPLIED
>

<!ELEMENT MOF:Namespace.contents (MOF:ModelElement | MOF:Feature |
    MOF:Namespace | MOF:Constraint |
    MOF:Import | MOF:TypedElement |
    MOF:Tag | MOF:StructuralFeature |
    MOF:BehavioralFeature |
    MOF:MofAttribute | MOF:Reference |
    MOF:Operation | MOF:MofException |
    MOF:GeneralizableElement |
    MOF:Classifier | MOF:Package |
    MOF:Class | MOF:Association |
    MOF:DataType | MOF:Parameter |
    MOF:Constant | MOF:AssociationEnd |
    MOF:TypeAlias)* >

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Mof.Model.Feature -->
<!-- _____ -->

<!ELEMENT MOF:Feature.visibility EMPTY >
<!ATTLIST MOF:Feature.visibility
    xmi.value (public_vis | protected_vis | private_vis) #REQUIRED
>

<!ELEMENT MOF:Feature.scope EMPTY >
<!ATTLIST MOF:Feature.scope
    xmi.value (instance_level | classifier_level) #REQUIRED
>

<!ELEMENT MOF:Feature (MOF:ModelElement.name |
    MOF:ModelElement.annotation |
    MOF:Feature.visibility | MOF:Feature.scope |
    XMI.extension | MOF:ModelElement.container |
    MOF:ModelElement.constraints)* >
<!ATTLIST MOF:Feature
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    visibility (public_vis | protected_vis | private_vis) #IMPLIED
    scope (instance_level | classifier_level) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

```

```

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.Classifier                                     -->
<!-- _____ -->

<!ELEMENT MOF:Classifier (MOF:ModelElement.name |
                        MOF:ModelElement.annotation |
                        MOF:GeneralizableElement.visibility |
                        MOF:GeneralizableElement.isAbstract |
                        MOF:GeneralizableElement.isRoot |
                        MOF:GeneralizableElement.isLeaf |
                        XMI.extension | MOF:ModelElement.container |
                        MOF:ModelElement.constraints |
                        MOF:GeneralizableElement.supertypes |
                        MOF:Namespace.contents)* >

<!ATTLIST MOF:Classifier
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    visibility (public_vis | protected_vis | private_vis) #IMPLIED
    isAbstract (true | false) #IMPLIED
    isRoot (yes | no | dont_care) #IMPLIED
    isLeaf (yes | no | dont_care) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    supertypes IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.Namespace                                     -->
<!-- _____ -->

<!ELEMENT MOF:Namespace (MOF:ModelElement.name |
                        MOF:ModelElement.annotation | XMI.extension |
                        MOF:ModelElement.container |
                        MOF:ModelElement.constraints |
                        MOF:Namespace.contents)* >

<!ATTLIST MOF:Namespace
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.StructuralFeature                                     -->
<!-- _____ -->

<!ELEMENT MOF:StructuralFeature.multiplicity (MOF:Multiplicity) >

```

```

<!ELEMENT MOF:StructuralFeature.isChangeable EMPTY >
<!ATTLIST MOF:StructuralFeature.isChangeable
          xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:StructuralFeature (MOF:ModelElement.name |
                                MOF:ModelElement.annotation |
                                MOF:Feature.visibility |
                                MOF:Feature.scope |
                                MOF:StructuralFeature.multiplicity |
                                MOF:StructuralFeature.isChangeable |
                                XMI.extension |
                                MOF:ModelElement.container |
                                MOF:ModelElement.constraints |
                                MOF:TypedElement.type)* >

<!ATTLIST MOF:StructuralFeature
          name CDATA #IMPLIED
          annotation CDATA #IMPLIED
          visibility (public_vis | protected_vis | private_vis) #IMPLIED
          scope (instance_level | classifier_level) #IMPLIED
          isChangeable (true | false) #IMPLIED
          container IDREFS #IMPLIED
          constraints IDREFS #IMPLIED
          type IDREFS #IMPLIED
          %XMI.element.att;
          %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Mof.Model.MofAttribute -->
<!-- _____ -->

<!ELEMENT MOF:MofAttribute.isDerived EMPTY >
<!ATTLIST MOF:MofAttribute.isDerived
          xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:MofAttribute (MOF:ModelElement.name |
                            MOF:ModelElement.annotation |
                            MOF:Feature.visibility | MOF:Feature.scope |
                            MOF:StructuralFeature.multiplicity |
                            MOF:StructuralFeature.isChangeable |
                            MOF:MofAttribute.isDerived | XMI.extension |
                            MOF:ModelElement.container |
                            MOF:ModelElement.constraints |
                            MOF:TypedElement.type)* >

<!ATTLIST MOF:MofAttribute
          name CDATA #IMPLIED
          annotation CDATA #IMPLIED
          visibility (public_vis | protected_vis | private_vis) #IMPLIED
          scope (instance_level | classifier_level) #IMPLIED
          isChangeable (true | false) #IMPLIED
          isDerived (true | false) #IMPLIED
          container IDREFS #IMPLIED

```

```

        constraints IDREFS #IMPLIED
        type IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.Operation                                     -->
<!-- _____ -->

<!ELEMENT MOF:Operation.isQuery EMPTY >
<!ATTLIST MOF:Operation.isQuery
            xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:Operation.exceptions (MOF:MofException)* >

<!ELEMENT MOF:Operation (MOF:ModelElement.name |
                        MOF:ModelElement.annotation |
                        MOF:Feature.visibility | MOF:Feature.scope |
                        MOF:Operation.isQuery | XMI.extension |
                        MOF:ModelElement.container |
                        MOF:ModelElement.constraints |
                        MOF:Operation.exceptions |
                        MOF:Namespace.contents)* >
<!ATTLIST MOF:Operation
        name CDATA #IMPLIED
        annotation CDATA #IMPLIED
        visibility (public_vis | protected_vis | private_vis) #IMPLIED
        scope (instance_level | classifier_level) #IMPLIED
        isQuery (true | false) #IMPLIED
        container IDREFS #IMPLIED
        constraints IDREFS #IMPLIED
        exceptions IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.Class                                     -->
<!-- _____ -->

<!ELEMENT MOF:Class.isSingleton EMPTY >
<!ATTLIST MOF:Class.isSingleton
            xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:Class (MOF:ModelElement.name |
                    MOF:ModelElement.annotation |
                    MOF:GeneralizableElement.visibility |
                    MOF:GeneralizableElement.isAbstract |
                    MOF:GeneralizableElement.isRoot |
                    MOF:GeneralizableElement.isLeaf |

```

```

MOF:Class.isSingleton | XMI.extension |
MOF:ModelElement.container |
MOF:ModelElement.constraints |
MOF:GeneralizableElement.supertypes |
MOF:Namespace.contents)* >
<!-- MOF:Class
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    visibility (public_vis | protected_vis | private_vis) #IMPLIED
    isAbstract (true | false) #IMPLIED
    isRoot (yes | no | dont_care) #IMPLIED
    isLeaf (yes | no | dont_care) #IMPLIED
    isSingleton (true | false) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    supertypes IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Mof.Model.ModelElement -->
<!-- _____ -->

<!-- MOF:ModelElement.name (#PCDATA | XMI.reference)* >

<!-- MOF:ModelElement.annotation (#PCDATA | XMI.reference)* >

<!-- MOF:ModelElement.container (MOF:Namespace)* >

<!-- MOF:ModelElement.constraints (MOF:Constraint)* >

<!-- MOF:ModelElement (MOF:ModelElement.name |
MOF:ModelElement.annotation | XMI.extension |
MOF:ModelElement.container |
MOF:ModelElement.constraints)* >
<!-- MOF:ModelElement
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Mof.Model.Package -->
<!-- _____ -->

<!-- MOF:Package (MOF:ModelElement.name |
MOF:ModelElement.annotation |
MOF:GeneralizableElement.visibility |
MOF:GeneralizableElement.isAbstract |

```

```

        MOF:GeneralizableElement.isRoot |
        MOF:GeneralizableElement.isLeaf | XMI.extension |
        MOF:ModelElement.container |
        MOF:ModelElement.constraints |
        MOF:GeneralizableElement.supertypes |
        MOF:Namespace.contents)* >
<!-- MOF:Package
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    visibility (public_vis | protected_vis | private_vis) #IMPLIED
    isAbstract (true | false) #IMPLIED
    isRoot (yes | no | dont_care) #IMPLIED
    isLeaf (yes | no | dont_care) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    supertypes IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- ----- -->
<!-- ----- -->
<!-- CLASS: Mof.Model.Constraint ----- -->
<!-- ----- -->

<!-- MOF:Constraint.expression (XMI.any)* >

<!-- MOF:Constraint.language (#PCDATA | XMI.reference)* >

<!-- MOF:Constraint.evaluationPolicy EMPTY >
<!-- MOF:Constraint.evaluationPolicy
    xmi.value (immediate | deferred) #REQUIRED
>

<!-- MOF:Constraint.constrainedElts (MOF:ModelElement)* >

<!-- MOF:Constraint (MOF:ModelElement.name |
    MOF:ModelElement.annotation |
    MOF:Constraint.expression |
    MOF:Constraint.language |
    MOF:Constraint.evaluationPolicy |
    XMI.extension | MOF:ModelElement.container |
    MOF:ModelElement.constraints |
    MOF:Constraint.constrainedElts)* >
<!-- MOF:Constraint
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    language CDATA #IMPLIED
    evaluationPolicy (immediate | deferred) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    constrainedElts IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

```

```

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.Association                                     -->
<!-- _____ -->

<!ELEMENT MOF:Association.isDerived EMPTY >
<!ATTLIST MOF:Association.isDerived
            xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:Association (MOF:ModelElement.name |
                           MOF:ModelElement.annotation |
                           MOF:GeneralizableElement.visibility |
                           MOF:GeneralizableElement.isAbstract |
                           MOF:GeneralizableElement.isRoot |
                           MOF:GeneralizableElement.isLeaf |
                           MOF:Association.isDerived | XMI.extension |
                           MOF:ModelElement.container |
                           MOF:ModelElement.constraints |
                           MOF:GeneralizableElement.supertypes |
                           MOF:Namespace.contents)* >
<!ATTLIST MOF:Association
            name CDATA #IMPLIED
            annotation CDATA #IMPLIED
            visibility (public_vis | protected_vis | private_vis) #IMPLIED
            isAbstract (true | false) #IMPLIED
            isRoot (yes | no | dont_care) #IMPLIED
            isLeaf (yes | no | dont_care) #IMPLIED
            isDerived (true | false) #IMPLIED
            container IDREFS #IMPLIED
            constraints IDREFS #IMPLIED
            supertypes IDREFS #IMPLIED
            %XMI.element.att;
            %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.Parameter                                     -->
<!-- _____ -->

<!ELEMENT MOF:Parameter.direction EMPTY >
<!ATTLIST MOF:Parameter.direction
            xmi.value (in_dir | out_dir | inout_dir | return_dir) #REQUIRED
>

<!ELEMENT MOF:Parameter.multiplicity (MOF:Multiplicity) >

<!ELEMENT MOF:Parameter (MOF:ModelElement.name |
                           MOF:ModelElement.annotation |
                           MOF:Parameter.direction |
                           MOF:Parameter.multiplicity | XMI.extension |
                           MOF:ModelElement.container |
                           MOF:ModelElement.constraints |

```

```

                                MOF:TypedElement.type)* >
<!-- MOF:Parameter
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    direction (in_dir | out_dir | inout_dir | return_dir) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    type IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
-->

<!-- CLASS: Mof.Model.BehavioralFeature
-->

<!-- MOF:BehavioralFeature (MOF:ModelElement.name |
                                MOF:ModelElement.annotation |
                                MOF:Feature.visibility |
                                MOF:Feature.scope | XMI.extension |
                                MOF:ModelElement.container |
                                MOF:ModelElement.constraints |
                                MOF:Namespace.contents)* >
<!-- MOF:BehavioralFeature
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    visibility (public_vis | protected_vis | private_vis) #IMPLIED
    scope (instance_level | classifier_level) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
-->

<!-- CLASS: Mof.Model.Import
-->

<!-- MOF:Import.visibility EMPTY >
<!-- MOF:Import.visibility
    xmi.value (public_vis | protected_vis | private_vis) #REQUIRED
-->

<!-- MOF:Import.importedNamespace (MOF:Namespace)* >

<!-- MOF:Import (MOF:ModelElement.name |
                                MOF:ModelElement.annotation |
                                MOF:Import.visibility | XMI.extension |
                                MOF:ModelElement.container |
                                MOF:ModelElement.constraints |
                                MOF:Import.importedNamespace)* >
<!-- MOF:Import
    name CDATA #IMPLIED

```

```

        annotation CDATA #IMPLIED
        visibility (public_vis | protected_vis | private_vis) #IMPLIED
        container IDREFS #IMPLIED
        constraints IDREFS #IMPLIED
        importedNamespace IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.GeneralizableElement -->
<!-- _____ -->

<!ELEMENT MOF:GeneralizableElement.visibility EMPTY >
<!ATTLIST MOF:GeneralizableElement.visibility
        xmi.value (public_vis | protected_vis | private_vis) #REQUIRED
>

<!ELEMENT MOF:GeneralizableElement.isAbstract EMPTY >
<!ATTLIST MOF:GeneralizableElement.isAbstract
        xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:GeneralizableElement.isRoot EMPTY >
<!ATTLIST MOF:GeneralizableElement.isRoot
        xmi.value (yes | no | dont_care) #REQUIRED
>

<!ELEMENT MOF:GeneralizableElement.isLeaf EMPTY >
<!ATTLIST MOF:GeneralizableElement.isLeaf
        xmi.value (yes | no | dont_care) #REQUIRED
>

<!ELEMENT MOF:GeneralizableElement.supertypes
                                                (MOF:GeneralizableElement)* >

<!ELEMENT MOF:GeneralizableElement (MOF:ModelElement.name |
        MOF:ModelElement.annotation |
        MOF:GeneralizableElement.visibility |
        MOF:GeneralizableElement.isAbstract |
        MOF:GeneralizableElement.isRoot |
        MOF:GeneralizableElement.isLeaf |
        XMI.extension |
        MOF:ModelElement.container |
        MOF:ModelElement.constraints |
        MOF:GeneralizableElement.supertypes |
        MOF:Namespace.contents)* >

<!ATTLIST MOF:GeneralizableElement
        name CDATA #IMPLIED
        annotation CDATA #IMPLIED
        visibility (public_vis | protected_vis | private_vis) #IMPLIED
        isAbstract (true | false) #IMPLIED
        isRoot (yes | no | dont_care) #IMPLIED
        isLeaf (yes | no | dont_care) #IMPLIED

```

```

        container IDREFS #IMPLIED
        constraints IDREFS #IMPLIED
        supertypes IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.Constant                                     -->
<!-- _____ -->

<!ELEMENT MOF:Constant.constValue (XMI.any) >

<!ELEMENT MOF:Constant (MOF:ModelElement.name |
                        MOF:ModelElement.annotation |
                        MOF:Constant.constValue | XMI.extension |
                        MOF:ModelElement.container |
                        MOF:ModelElement.constraints |
                        MOF:TypedElement.type)* >
<!ATTLIST MOF:Constant
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    type IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.Reference                                     -->
<!-- _____ -->

<!ELEMENT MOF:Reference.referencedEnd (MOF:AssociationEnd)* >

<!ELEMENT MOF:Reference (MOF:ModelElement.name |
                        MOF:ModelElement.annotation |
                        MOF:Feature.visibility | MOF:Feature.scope |
                        MOF:StructuralFeature.multiplicity |
                        MOF:StructuralFeature.isChangeable |
                        XMI.extension | MOF:ModelElement.container |
                        MOF:ModelElement.constraints |
                        MOF:TypedElement.type |
                        MOF:Reference.referencedEnd)* >
<!ATTLIST MOF:Reference
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    visibility (public_vis | protected_vis | private_vis) #IMPLIED
    scope (instance_level | classifier_level) #IMPLIED
    isChangeable (true | false) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    type IDREFS #IMPLIED

```

```

        referencedEnd IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.MofException                                     -->
<!-- _____ -->

<!ELEMENT MOF:MofException (MOF:ModelElement.name |
                            MOF:ModelElement.annotation |
                            MOF:Feature.visibility | MOF:Feature.scope |
                            XMI.extension | MOF:ModelElement.container |
                            MOF:ModelElement.constraints |
                            MOF:Namespace.contents)* >

<!ATTLIST MOF:MofException
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    visibility (public_vis | protected_vis | private_vis) #IMPLIED
    scope (instance_level | classifier_level) #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.DataType                                     -->
<!-- _____ -->

<!ELEMENT MOF:DataType.typeCode ANY >

<!ELEMENT MOF:DataType (MOF:ModelElement.name |
                        MOF:ModelElement.annotation |
                        MOF:GeneralizableElement.visibility |
                        MOF:GeneralizableElement.isAbstract |
                        MOF:GeneralizableElement.isRoot |
                        MOF:GeneralizableElement.isLeaf |
                        MOF:DataType.typeCode | XMI.extension |
                        MOF:ModelElement.container |
                        MOF:ModelElement.constraints |
                        MOF:GeneralizableElement.supertypes |
                        MOF:Namespace.contents)* >

<!ATTLIST MOF:DataType
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    visibility (public_vis | protected_vis | private_vis) #IMPLIED
    isAbstract (true | false) #IMPLIED
    isRoot (yes | no | dont_care) #IMPLIED
    isLeaf (yes | no | dont_care) #IMPLIED
    typeCode CDATA #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED

```

```

        supertypes IDREFS #IMPLIED
        %XMI.element.att;
        %XMI.link.att;
    >

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.AssociationEnd                                     -->
<!-- _____ -->

<!ELEMENT MOF:AssociationEnd.multiplicity (MOF:Multiplicity) >

<!ELEMENT MOF:AssociationEnd.aggregation EMPTY >
<!ATTLIST MOF:AssociationEnd.aggregation
            xmi.value (none | shared | composite) #REQUIRED
>

<!ELEMENT MOF:AssociationEnd.isNavigable EMPTY >
<!ATTLIST MOF:AssociationEnd.isNavigable
            xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:AssociationEnd.isChangeable EMPTY >
<!ATTLIST MOF:AssociationEnd.isChangeable
            xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:AssociationEnd.otherEnd (MOF:AssociationEnd) >

<!ELEMENT MOF:AssociationEnd (MOF:ModelElement.name |
                                MOF:ModelElement.annotation |
                                MOF:AssociationEnd.multiplicity |
                                MOF:AssociationEnd.aggregation |
                                MOF:AssociationEnd.isNavigable |
                                MOF:AssociationEnd.isChangeable |
                                MOF:AssociationEnd.otherEnd |
                                XMI:extension | MOF:ModelElement.container |
                                MOF:ModelElement.constraints |
                                MOF:TypedElement.type)* >

<!ATTLIST MOF:AssociationEnd
            name CDATA #IMPLIED
            annotation CDATA #IMPLIED
            aggregation (none | shared | composite) #IMPLIED
            isNavigable (true | false) #IMPLIED
            isChangeable (true | false) #IMPLIED
            container IDREFS #IMPLIED
            constraints IDREFS #IMPLIED
            type IDREFS #IMPLIED
            %XMI.element.att;
            %XMI.link.att;
>

<!-- _____ -->
<!--                                     -->
<!-- CLASS: Mof.Model.TypedElement                                     -->

```

```

<!-- _____ -->

<!ELEMENT MOF:TypedElement.type (MOF:Classifier)* >

<!ELEMENT MOF:TypedElement (MOF:ModelElement.name |
                                MOF:ModelElement.annotation | XMI.extension |
                                MOF:ModelElement.container |
                                MOF:ModelElement.constraints |
                                MOF:TypedElement.type)* >

<!ATTLIST MOF:TypedElement
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    type IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Mof.Model.Multiplicity -->
<!-- _____ -->

<!ELEMENT MOF:Multiplicity.lower (#PCDATA | XMI.reference)* >

<!ELEMENT MOF:Multiplicity.upper (#PCDATA | XMI.reference)* >

<!ELEMENT MOF:Multiplicity.isOrdered EMPTY >
<!ATTLIST MOF:Multiplicity.isOrdered
    xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:Multiplicity.isUnique EMPTY >
<!ATTLIST MOF:Multiplicity.isUnique
    xmi.value (true | false) #REQUIRED
>

<!ELEMENT MOF:Multiplicity (MOF:Multiplicity.lower |
                                MOF:Multiplicity.upper |
                                MOF:Multiplicity.isOrdered |
                                MOF:Multiplicity.isUnique | XMI.extension)* >

<!ATTLIST MOF:Multiplicity
    lower CDATA #IMPLIED
    upper CDATA #IMPLIED
    isOrdered (true | false) #IMPLIED
    isUnique (true | false) #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Mof.Model.TypeAlias -->
<!-- _____ -->

```

```

<!ELEMENT MOF:TypeAlias.multiplicity (MOF:Multiplicity) >

<!ELEMENT MOF:TypeAlias (MOF:ModelElement.name |
                        MOF:ModelElement.annotation |
                        MOF:TypeAlias.multiplicity | XMI.extension |
                        MOF:ModelElement.container |
                        MOF:ModelElement.constraints |
                        MOF:TypedElement.type)* >

<!ATTLIST MOF:TypeAlias
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    type IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!-- _____ -->
<!-- _____ -->
<!-- CLASS: Mof.Model.Tag -->
<!-- _____ -->

<!ELEMENT MOF:Tag.tagId (#PCDATA | XMI.reference)* >

<!ELEMENT MOF:Tag.values (XMI.any) >

<!ELEMENT MOF:Tag.elements (MOF:ModelElement)* >

<!ELEMENT MOF:Tag (MOF:ModelElement.name | MOF:ModelElement.annotation |
                    MOF:Tag.tagId | MOF:Tag.values | XMI.extension |
                    MOF:ModelElement.container |
                    MOF:ModelElement.constraints | MOF:Tag.elements)* >

<!ATTLIST MOF:Tag
    name CDATA #IMPLIED
    annotation CDATA #IMPLIED
    tagId CDATA #IMPLIED
    container IDREFS #IMPLIED
    constraints IDREFS #IMPLIED
    elements IDREFS #IMPLIED
    %XMI.element.att;
    %XMI.link.att;
>

<!ELEMENT MOF:Mof (MOF:Model)* >
<!ATTLIST MOF:Mof
    %XMI.element.att;
    %XMI.link.att;
>

<!ELEMENT MOF:Model (MOF:Feature | MOF:Classifier | MOF:Namespace |
                    MOF:StructuralFeature | MOF:MofAttribute |
                    MOF:Operation | MOF:Class | MOF:ModelElement |
                    MOF:Package | MOF:Constraint | MOF:Association |

```

```
MOF:Parameter | MOF:BehavioralFeature | MOF:Import |  
MOF:GeneralizableElement | MOF:Constant |  
MOF:Reference | MOF:MofException | MOF:DataType |  
MOF:AssociationEnd | MOF:TypedElement |  
MOF:Multiplicity | MOF:TypeAlias | MOF:Tag)* >  
<!--ATTLIST MOF:Model  
    %XMI.element.att;  
    %XMI.link.att;  
>
```

Example Model Encodings

C

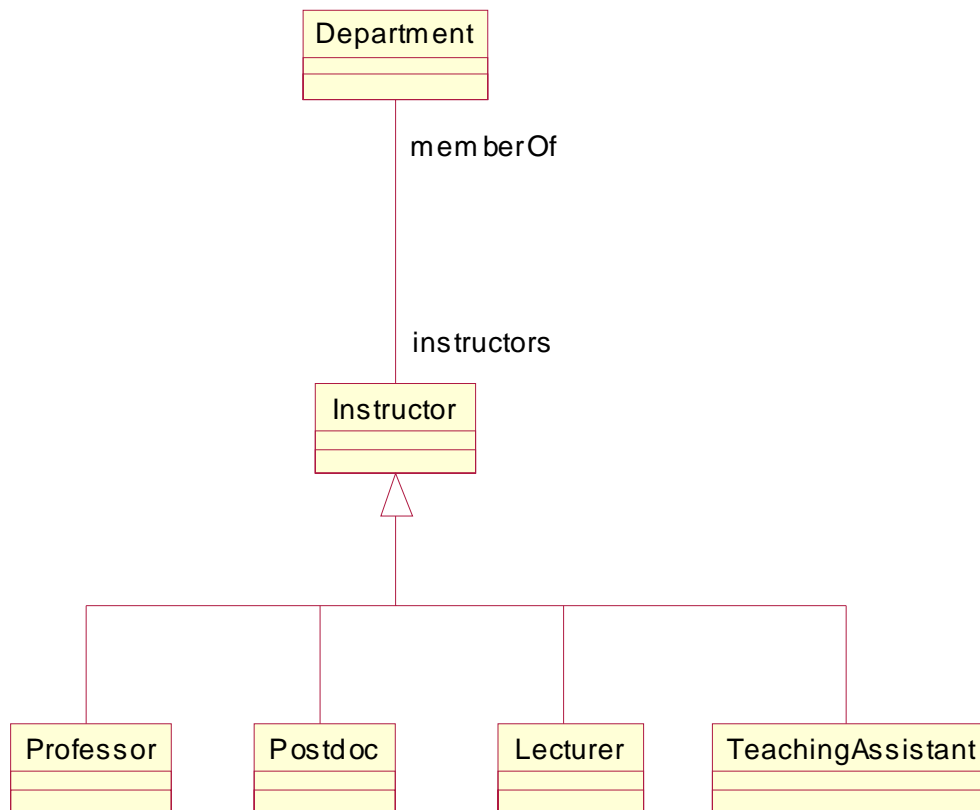
C.1 Introduction

This appendix provides the following example UML models and their resultant XML encoding.

- An example of a department at a university.
- An example of a model of a schedule, using the department and a campus model.
- An example of instances of a schedule.
- An example using an IDL datatype metamodel for “complete” datatype metamodels.

C.2 Example 1 - Department

This is an example of a model of a department, using the UML DTD specified in Appendix A



The Department class has a relationship called **instructors** to **Instructors**, a superclass for **Professor**, **Postdoc**, **Lecturer**, and **Teaching Assistant**.

```

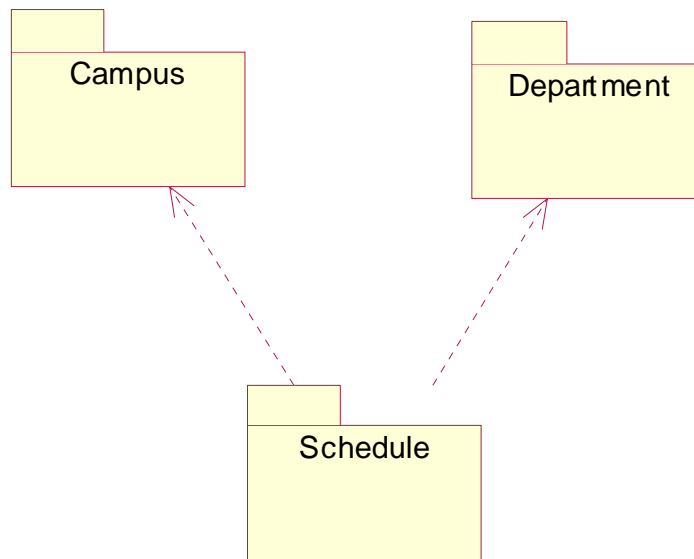
File="Department.xml" Namespace="Department":
<XMI version="1.1" xmlns:UML="org.omg/UML1.3">
  <XMI.header>
    <XMI.model xmi.name="Department" href="Department.xml"/>
    <XMI.metamodel xmi.name="UML" href="UML.xml"/>
  </XMI.header>
  <XMI.content>
    <UML:Class name="Department" xmi.id="Department"/>
    <UML:Class name="Instructor" xmi.id="Instructor"/>
    <UML:Class name="Professor" xmi.id="Professor" generalization="Instructor"/>
    <UML:Class name="Postdoc" xmi.id="Postdoc" generalization="Instructor"/>
    <UML:Class name="Lecturer" xmi.id="Lecturer" generalization="Instructor"/>
    <UML:Class name="TeachingAssistant" xmi.id="TeachingAssistant" generalization="Instructor"/>
    <UML:Association>
      <UML:Association.connection>
        <UML:AssociationEnd name="instructors" type="Instructor"/>
        <UML:AssociationEnd name="memberOf" type="Department"/>
      </UML:Association.connection>
    </UML:Association>
  </XMI.content>
</XMI>

```

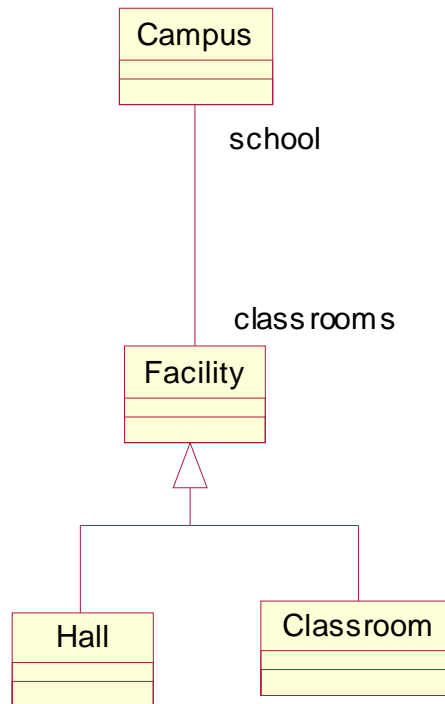
In the above example, the namespace of UML is used in the namespace declaration in the XMI element as well as the XMI.metamodel tag. The model's name is Department, which will also serve as its namespace in example 3. Each class is listed in the content, including superclass references. For simplicity in the example, the ids were chosen to be the same as the name. The association between Instructor and Department is also shown.

C.3 Example 2 - Campus and Schedule

This example contains three models, the department above, a campus, and a schedule which uses both models.



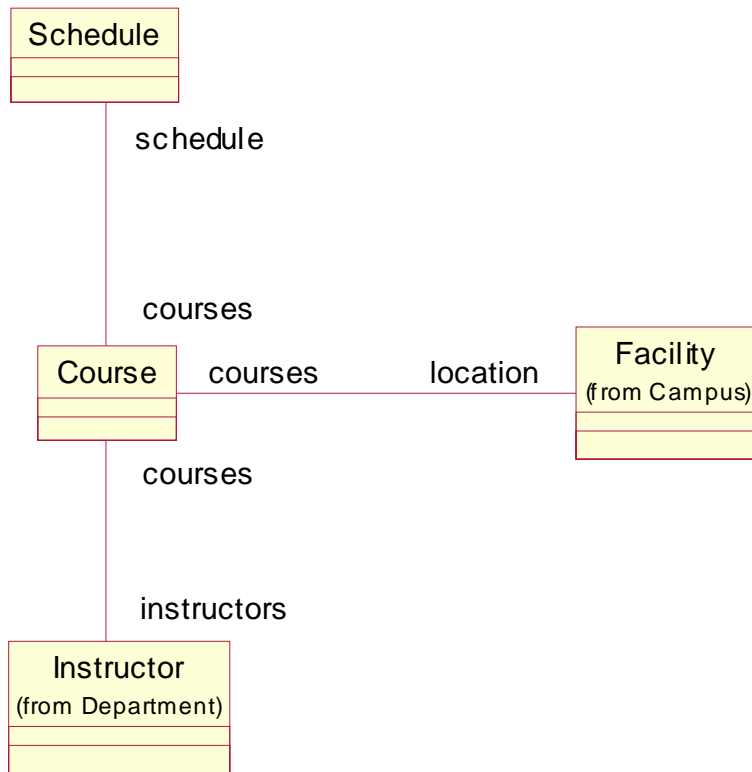
The Campus model has a Campus associated with Facilities, which have subclasses of Hall and Classroom. The Campus and Department models are not related, but the Schedule model uses both.



```

File="Campus.xml" Namespace="Campus":
<XMI version="1.1" xmlns:UML="org.omg/UML1.3">
  <XMI.header>
    <XMI.model xmi.name="Campus" href="Campus.xml"/>
    <XMI.metamodel xmi.name="UML" href="UML.xml"/>
  </XMI.header>
  <XMI.content>
    <UML:Class name="Campus" xmi.id="Campus"/>
    <UML:Class name="Facility" xmi.id="Facility"/>
    <UML:Class name="Classroom" xmi.id="Classroom" generalization="Facility"/>
    <UML:Class name="Hall" xmi.id="Hall" generalization="Facility"/>
    <UML:Association>
      <UML:Association.connection>
        <UML:AssociationEnd name="classrooms" type="Facility"/>
        <UML:AssociationEnd name="school" type="Campus"/>
      </UML:Association.connection>
    </UML:Association>
  </XMI.content>
</XMI>

```



The schedule model has a Schedule based on Courses, where each Course is associated with a Facility from the Campus model and an Instructor from the Department model.

```
File="Schedule.xml" Namespace="Schedule":
<XMI version="1.1" xmlns:UML="org.omg/UML1.3">
  <XMI.header>
    <XMI.model xmi.name="Schedule" href="Schedule.xml"/>
    <XMI.metamodel xmi.name="UML" href="UML.xml"/>
    <XMI.import name="Department" href="Department.xml"/>
    <XMI.import name="Campus" href="Campus.xml"/>
  </XMI.header>
  <XMI.content>
    <UML:Class name="Schedule" xmi.id="Schedule"/>
    <UML:Class name="Course" xmi.id="Course"/>
    <UML:Association>
      <UML:Association.connection>
        <UML:AssociationEnd name="courses" type="Course"/>
        <UML:AssociationEnd name="schedule" type="Schedule"/>
      </UML:Association.connection>
    </UML:Association>
    <UML:Association>
      <UML:Association.connection>
        <UML:AssociationEnd name="instructors">
          <UML:AssociationEnd.type>
            <UML:Classifier href="Department.xml#Instructor"/>
          </UML:AssociationEnd.type>
        </UML:AssociationEnd>
        <UML:AssociationEnd name="courses" type="Course"/>
      </UML:Association.connection>
    </UML:Association>
    <UML:Association>
      <UML:Association.connection>
        <UML:AssociationEnd name="location">
          <UML:AssociationEnd.type>
            <UML:Classifier href="Campus.xml#Facility"/>
          </UML:AssociationEnd.type>
        </UML:AssociationEnd>
        <UML:AssociationEnd name="courses" type="Course"/>
      </UML:Association.connection>
    </UML:Association>
  </XMI.content>
</XMI>
```

Since the Campus and Department classes are in different files, XMI.import statements are generated as well as cross-document linking proxies for the AssociationEnd's type.

C.4 Example 3 - Instances of Schedule

In this example, each of the models are instantiated. The models are referred to using the XMI.metamodel tags and the namespace declarations are used in the XMI element.

Chemistry is an instance of department:

```

File="Chemistry.xml" Namespace="Chemistry":
<XMI version="1.1" xmlns:Department="edu.university/Department">
  <XMI.header>
    <XMI.model xmi.name="Chemistry" href="Chemistry.xml"/>
    <XMI.metamodel xmi.name="Department" href="Department.xml"/>
  </XMI.header>
  <XMI.content>
    <Department:Department name="Chemistry">
      <Department:Department.instructors>
        <Department:Professor name="Dr. Smith" xmi.id="Smith"/>
        <Department:Postdoc name="Dr. Jones" xmi.id="Jones"/>
        <Department:Lecturer name="Dr. Visitor" xmi.id="Visitor"/>
        <Department:TeachingAssistant name="Fred" xmi.id="Fred"/>
      </Department:Department.instructors>
    </Department:Department>
  </XMI.content>
</XMI>

```

The Department is considered the metamodel for Chemistry. The Department namespace is used and is listed in XMI.metamodel. This example shows Chemistry is a Department with four instances of Instructor's subclasses, Professor Smith, Postdoc Jones, Lecturer Visitor, and TeachingAssistant Fred.

UCLA is an instance of the Campus model. The Campus namespace is used and is listed in XMI.metamodel. UCLA is a Campus with three instances of Facility's subclasses, Classroom 101, Classroom 202, and Hall Branner Hall.

```

File="UCLA.xml" Namespace="UCLA":
<XMI version="1.1" xmlns:Campus="edu.university/Campus">
  <XMI.header>
    <XMI.model xmi.name="UCLA" href="UCLA.xml"/>
    <XMI.metamodel xmi.name="Campus" href="Campus.xml"/>
  </XMI.header>
  <XMI.content>
    <Campus:Campus name="UCLA">
      <Campus:Campus.classrooms>
        <Campus:Classroom name="Room 101" xmi.id="101" capacity="30"/>
        <Campus:Classroom name="Room 202" xmi.id="202" capacity="30"/>
        <Campus:Hall name="Branner Hall" xmi.id="BH" capacity="100"/>
      </Campus:Campus.classrooms>
    </Campus:Campus>
  </XMI.content>
</XMI>

```

Spring2000 is an instance of Schedule that uses UCLA, an instance of Campus, and Chemistry, an instance of Department. The Schedule namespace is used and is listed in XMI.metamodel. Also, Schedule has imported Campus and Department as declared in the XMI.import elements. Schedule and Campus are also used as metamodels for Spring2000, and are also declared in the XMI element and in the XMI.metamodel elements. Spring2000 uses UCLA and Chemistry, as listed in the XMI.import elements.

```

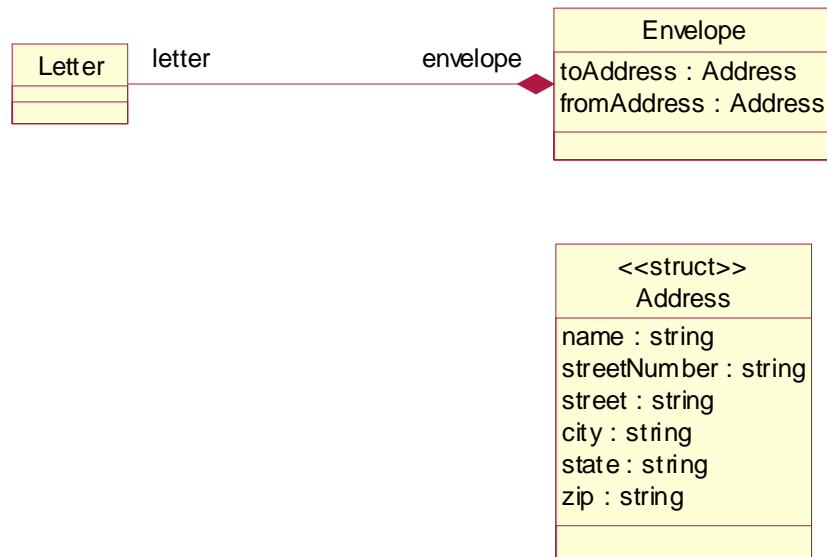
File="Spring2000.xml" Namespace="Spring2000":
<XMI version="1.1"
  xmlns:Schedule="edu.university/Schedule"
  xmlns:Campus="edu.university/Campus"
  xmlns:Department="edu.university/Department">
  <XMI.header>
    <XMI.model xmi.name="Spring2000" href="Spring2000.xml"/>
    <XMI.metamodel xmi.name="Schedule" href="Schedule.xml"/>
    <XMI.metamodel xmi.name="Department" href="Department.xml"/>
    <XMI.metamodel xmi.name="Campus" href="Campus.xml"/>
    <XMI.import name="Chemistry" href="Chemistry.xml"/>
    <XMI.import name="UCLA" href="UCLA.xml"/>
  </XMI.header>
  <XMI.content>
    <Schedule:Schedule name="Spring 2000">
      <Schedule:Schedule.courses>
        <Schedule:Course name="Chemistry 1" xmi.id="Chem1">
          <Schedule:Course.instructors>
            <Department:Professor href="Chemistry.xml#Smith"/>
            <Department:TeachingAssistant href="Chemistry.xml#Fred"/>
          </Schedule:Course.instructors>
          <Schedule:Course.location>
            <Campus:Hall href="UCLA.xml#BH"/>
          </Schedule:Course.location>
        </Schedule:Course>
      </Schedule:Schedule.courses>
    </Schedule:Schedule>
  </XMI.content>
</XMI>

```

Spring2000 is an instance of Schedule which has a Course instance Chemistry 1 with instructors Professor Smith and TeachingAssistant Fred from the Department instance in Chemistry.xml and location Hall Branner Hall from the Campus instance in UCLA.xml.

C.5 Example 4 - “Complete” datatype metamodel.

As an example, suppose we have a model called Mail of letters and envelopes. The model consists of two classes called Letter and Envelope, where Envelope contains Letter. Envelope has two attributes, toAddress and fromAddress, both of type Address. Address is an IDL struct with fields name, streetNumber, street, city, state, and zip, all of IDL type string.



As an example of the standard IDL metamodel, we will use the Base IDL metamodel from the Corba Components Model (CCM) specification, orbos/99-07-02, Volume 2, Chapter 2.4.1. The Base IDL metamodel should subclass MOF:DataType. The following is the declaration of StructDef directly from the submission.

```

<!-- _____ -->
<!-- -->
<!-- METAMODEL CLASS: BaseIDL.StructDef -->
<!-- -->
<!-- _____ -->

<!ELEMENT BaseIDL.StructDef.members (BaseIDL.Field)>
<!ENTITY % BaseIDL.StructDefProperties
'(%BaseIDL.TypedefDefProperties;
, (BaseIDL.StructDef.members)*)' >
<!ENTITY % BaseIDL.StructDefAssociations
'(%BaseIDL.TypedefDefAssociations;)' >
<!ELEMENT BaseIDL.StructDef ( %BaseIDL.StructDefProperties;
, (XMI.extension* , %BaseIDL.StructDefAssociations; ) )?>
<!ATTLIST BaseIDL.StructDef %XMI.element.att; %XMI.link.att; >
  
```

Since Address is a definition of a struct, it is an instance of the IDL metaclass StructDef.

The XMI document for the Mail model uses the UML DTD for Letter and Envelope and the IDL DTD for Address:

```

<XMI version="1.1" xmlns:UML="omg.org/UML">
  <XMI.header>
    <XMI.model xmi.name="Mail" href="mail.xml"/>
    <XMI.metamodel xmi.name="UML" href="uml.xml"/>
    <XMI.metamodel xmi.name="IDL" href="idl.xml"/>
  </XMI.header>
  <XMI.content>
    <UML:Class name="Letter" xmi.id="idLetter"/>
    <UML:Class name="Envelope" xmi.id="idEnvelope">
      <UML:ModelElement.ownedElements>
        <UML:Attribute name="toAddress" type="idAddress" />
        <UML:Attribute name="fromAddress" type="idAddress" />
      </UML:ModelElement.ownedElements>
    </UML:Class>
    <UML:Association>
      <UML:Association.connections>
        <UML:AssociationEnd xmi.id="idenvelope" isComposite="true"/>
        <UML:AssociationEnd xmi.id="idletter"/>
      </UML:Association.connections>
    </UML:Association>

    <BaseIDL.StructDef xmi.id="idAddress">
      <BaseIDL.Contained.identifier>Address</BaseIDL.Contained.identifier>
      <BaseIDL.StructDef.members>
        <BaseIDL.Field>
          <BaseIDL.Field.identifier>name</BaseIDL.Field.identifier>
          <BaseIDL.Typed.idlType> <BaseIDL.StringDef/> </BaseIDL.Typed.idlType>
        </BaseIDL.Field>
        <BaseIDL.Field>
          <BaseIDL.Field.identifier>streetNumber</BaseIDL.Field.identifier>
          <BaseIDL.Typed.idlType> <BaseIDL.StringDef/> </BaseIDL.Typed.idlType>
        </BaseIDL.Field>
        <BaseIDL.Field>
          <BaseIDL.Field.identifier>street</BaseIDL.Field.identifier>
          <BaseIDL.Typed.idlType> <BaseIDL.StringDef/> </BaseIDL.Typed.idlType>
        </BaseIDL.Field>
        <BaseIDL.Field>
          <BaseIDL.Field.identifier>city</BaseIDL.Field.identifier>
          <BaseIDL.Typed.idlType> <BaseIDL.StringDef/> </BaseIDL.Typed.idlType>
        </BaseIDL.Field>
        <BaseIDL.Field>
          <BaseIDL.Field.identifier>state</BaseIDL.Field.identifier>
          <BaseIDL.Typed.idlType> <BaseIDL.StringDef/> </BaseIDL.Typed.idlType>
        </BaseIDL.Field>
        <BaseIDL.Field>
          <BaseIDL.Field.identifier>zip</BaseIDL.Field.identifier>
          <BaseIDL.Typed.idlType> <BaseIDL.StringDef/> </BaseIDL.Typed.idlType>
        </BaseIDL.Field>
      </BaseIDL.StructDef.members>
    </BaseIDL.StructDef>

  </XMI.content>
</XMI>

```

The DTD that would be generated for the Mail model is:

```

<ELEMENT Letter (Letter.envelope | XMI.extension)*>
<ATTLIST Letter envelope IDREF #IMPLIED %XMI.element.att; %XMI.link.att; >
<ELEMENT Letter.envelope (Envelope | XMI.extension)*>

<ELEMENT Envelope (Envelope.letter |
                    Envelope.toAddress |
                    Envelope.fromAddress |
                    XMI.extension)*>
<ATTLIST Envelope letter IDREF #IMPLIED %XMI.element.att; %XMI.link.att; >
<ELEMENT Envelope.toAddress (Address | XMI.extension)*>
<ELEMENT Envelope.fromAddress (Address | XMI.extension)*>

<ELEMENT Address (Address.name | Address.streetName | Address.street |
                  Address.city | Address.state | Address.zip |
                  XMI.extension)*>
<ATTLIST Address name CDATA #IMPLIED
                streetNumber CDATA #IMPLIED
                street CDATA #IMPLIED
                city CDATA #IMPLIED
                state CDATA #IMPLIED
                zip CDATA #IMPLIED
                %XMI.element.att; %XMI.link.att; >
<ELEMENT Address.name (PCDATA | XMI.extension)*>
<ELEMENT Address.streetNumber (PCDATA | XMI.extension)*>
<ELEMENT Address.street (PCDATA | XMI.extension)*>
<ELEMENT Address.city (PCDATA | XMI.extension)*>
<ELEMENT Address.state (PCDATA | XMI.extension)*>
<ELEMENT Address.zip (PCDATA | XMI.extension)*>

```

An XMI document instance of the Mail model is:

```

<XMI version="1.1">
  <XMI.header>
    <XMI.model xmi.name="myMail" href="myMail.xml"/>
    <XMI.metamodel xmi.name="Mail" href="mail.xml"/>
  </XMI.header>
  <XMI.content>

    <Envelope xmi.id="myEnvelope" letter="myLetter">
      <Envelope.toAddress name="Sridhar" streetNumber="25725"
                        street="Jeronimo" city="Mission Viejo" state="CA" zip="92691" />
      <Envelope.fromAddress name="Steve" streetNumber="555"
                        street="Bailey" city="San Jose" state="CA" zip="95141" />
    </Envelope>
    <Letter xmi.id="myLetter" envelope="myEnvelope"/>

  </XMI.content>
</XMI>

```

