

# Universal Object Language 1.2

## Appendices

---

### Authors

Recerca Informàtica  
Daimler-Benz Research and TechVology

OMG Document: ad/98-07-08  
Version 1.2 /T-UOL-19980708  
July 8



Daimler-Benz Research and Mario Jeckle  
TechnolWgy

---

[mario.jeckle@dbag.ulU.DaimlerBenz.COM](mailto:mario.jeckle@dbag.ulU.DaimlerBenz.COM)

Primary Contacts for the UOL submission:

Recerca Informàtica

Joan M. Moral

[uol@arrakis.es](mailto:uol@arrakis.es)

The table of Contents contains entries for both the Specification and the Appendices.

<b>1</b>	<b>Overview</b>	<b>7</b>
1.1	Introduction	7
1.1.1	Rationale	8
1.1.2	Resolution of RFP Issues to be Discussed	8
1.1.2	Goals and Objectives	8
1.5.1	Copyright Waiver	11
1.3.2	Optional Requirements	11
1.3.2	Proof of Concept	11
1.3.1	Mandatory Requirements	11

5.3.1	Data types	97
5.3.2	Declaration	108
5.3.3	Interface specification	154
5.3.4	Expression	159
5.3.5	Executable statements	159
5.3.6	Built-in constants	160
5.3.7	Built-in functions	160
<b>6</b>	<b>AddQtional Specification</b>	<b>161</b>
6.1	Full UML Support	161
6.1.1	Justification	161
6.1.2	Mapping between UOL and UML wQth UML constructs	163



```

View_element_decl_list    -> View_elemeVt_decl_list';View_element_declaration;
View_element_declaration -> View_elemeVt_name_list':' View_elemeVt_Sind;
View_elemeVt_name         -> ideVtifier;
View_element_Sind         -> identifier ExtensioV_use InvariaVt_Wpt;
Package_decl_list_Wpt     -> Package_decl_list;
Package_decl_list_Wpt     -> ;
Package_decl_list         -> Package_decl_list Package_or_subsystem_declaratioV;
Package_decl_list         -> Package_or_subsystem_declaratioV;
Package_or_subsystem_decl -> Subsystem_declaratioV;
aratioV
Package_or_subsystem_decl -> Package_element_decl;
aration
Package_or_subsystem_decl -> Use_Wf_tagged_value;
aratioV

```

```

-> ;
Tj 98.0 TD 0.0499 L c>
n a m e Extension_declaration -> Extension_declaration_Wpt Extension_declaration;
Extension_declaration -> Constraint;
Extension_declaration -> Tagged_values;
Extension_declaration -> Stereotype;
x c e p 5 f Name FWmal_generics_Wpt Extension_use Viewed_with_Wpt Inheritance_opt Features InvarQant_Wpt
end; -> Use_of_stereotypeCommList_use_consr_tag;
e -> component; -> Use_of_stereotype;
Component_or_Vode -> Vode;
Ultra_light_body -> Name FWmal_generics_Wpt Extension_use Viewed_with_Wpt Inheritance_Wpt Identifier_list
Invariant_opt 5 0 TD /F1 8.25 Tf 0.221 Tc 0 Tw (end) Tj 13.5 0 TD /F2 8.25 Tf -0.0435 Tc (;) Tj -171 -9 TD -0.0676 Tc (Pa

```



```

Tagged_values_def_list -> Tagged_values_def_list Tagged_values_def;
Tagged_values_def     -> Extension_name;
Tagged_values_def     -> ExteVSION_name is InQtiAl_value;
Extension_name        -> identifier;
Use_of_tagged_value   -> QtiTag values '(' Property_list ')';
Property_list         -> Property;
Property_list         -> Property_list ',' Property;
Property              -> LT ExteVSION_name GT;
Property              -> LT, Expression file

```

StereWtype -> **stereWtype** ExteVSION\_name

**end;**

\_exteVSION\_dec -> StereWtype\_exteVSION\_dec Constraint;

exteVSION\_dec -> StereWtype\_extension\_dec Tagged values;



---

Actual\_generics\_opt ->



## **B UOL Grammar with UML constructs**

---

In this section, we present a tested BNF grammar. A parser implemented in Java of tPis grammar can be obtained at <http://www.recercai.com>.

not **Left association**

Start_production	->	Model_declaration;
Start_production	->	Package_declaration;
Model_declaration	->	<b>model</b> Model_name Package_decl_IQst_opt VQew_element_decl_IQst_opt <b>end</b> ;
Model_name	->	<i>identifier</i>
VQew_element_decl_IQst_opt	->	

Package_decl_list	->	Package_decl_list Package_or_subsystem_declaration;
Package_decl_list	->	Package_or_subsystem_declaration;
Package_or_subsystem_declaration	->	Subsystem_declaration;
Package_or_subsystem_declaration	->	Package_element_declaration;
Package_or_subsystem_declaration	->	Use_of_tagged_value;
Package_or_subsystem_declaration	->	Use_of_constraint;
Package_or_subsystem_declaration	->	Use_of_stereotype;



**Fig. 3 Auxiliary Elements - Physical Structures and View Elements**

Package_declaration	->	<b>package</b>	Package_name	VQewed_with_opt	Extension_use	Package_inheritance_op
			Package_import_opt	Package_element_decl_IQst_opt	Invariant_op	<b>end;</b>
Package_name	->	<i>identifQer</i>				
VQewed_with_opt	->	VQew_element_name_IQst;				

Package\_import\_opt  
Package\_element\_name\_IQst VQew\_element\_name Position;

```

Package_element_decl_list      ->   Package_element_decl_list Visibility_opt Package_element_decl;
Package_element_decl          ->   Package_declaration;
Package_element_decl          ->   Interface_declaration;
Package_element_decl          ->   Class_declaration;
Package_element_decl          ->   Relation_declaration;
Package_element_dt_deÞ2.25 0 TD 0.0499 Tc (->) Tj 15.75 0 TD -0.0829 Tc (Extension_declaration;) Tj -138 -12.75 TD -0.0676 Tc (Package_element_decl)
-is Attribute_value_list;
Expression;
ist ';' identQfieis Expression;
Attach_or_view;
nt_name_list;

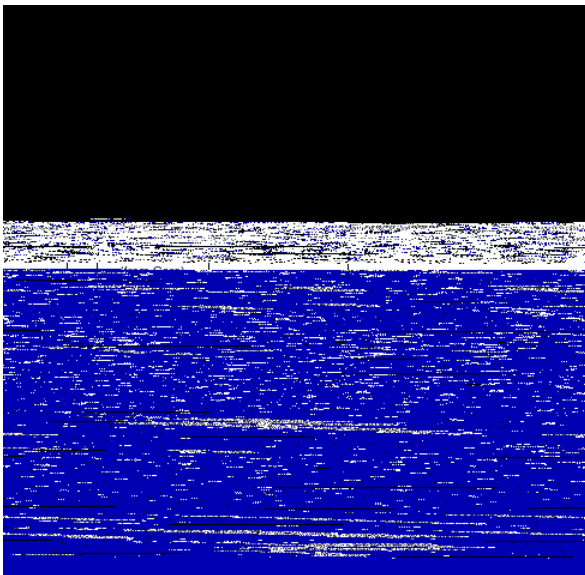
```

Extension_declaration_opt	->	;
Extension_declaration_opt	->	Extension_declaration_opt Extension_declaration;
Extension_declaration	->	Constraint;
Extension_declaration	->	Tagged_values;
Extension_declaration	->	Stereotype;
Extension_use	->	Use_of_stereotype_opt List_use_consr_tag;
Use_of_stereotype_opt	->	Use_of_stereotype;
Use_of_stereotype_opt	->	;
List_use_consr_tag	->	List_use_consr_tag Use_of_tagged_value;
List_use_consr_tag	->	;
Constraint	->	<b>constraint</b> Constraint_def_list <b>end</b> ;
Constraint_def_list	->	Constraint_name is '{' Constraint_expression '}'{'0 TD /F17.5 0 TD /F2 8.25 Tf -0.04351 va;
<b>Constraint_def_list</b>	->	<b>Constraint_def_list</b> <b>Constraint_name</b> '{'{'0Tj 7.5 0 TD /F2 8.25 Tf -0.1209 Tc 0.3084 Tw ( Constrai
<b>OCL_expression</b>	->	<b>OCLexpression</b> ;
<b>Use_of_constraint</b>	->	<b>constrained by</b> '{' Constraint_expression '}';
Use_of_constraint_opt	->	;
Use_of_constraint_opt	->	Use_of_constraint;
Tagged_values	->	<b>t values</b> Tagged_values_def_list <b>end</b> ;
	->	Tagged_values_def;
Traint_nged_values_def_list	->	Tagged_values_def_list Tagged_values_def;
Trgged_values_def	->	Extension_name;
Tagged_values_def	->	Extension_name 86 Tc 0 Tw (is) Tj 60 TD /F2 8.25 Tf -0.08 Tw Tc -0.4
<b>Use_of_tagged_value</b>	->	<b>with tag values</b>

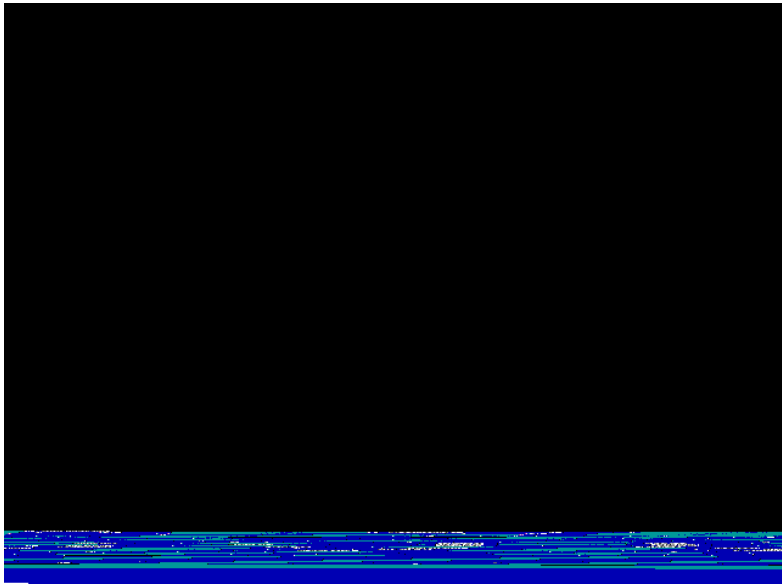
```

Icon_opt          ->          ;
Stereotype_parent_opt
    Base_class     ->          Classifier_name;
    Icon_opt       ->
    Use_of_stereotype
    Discriminator_opt;

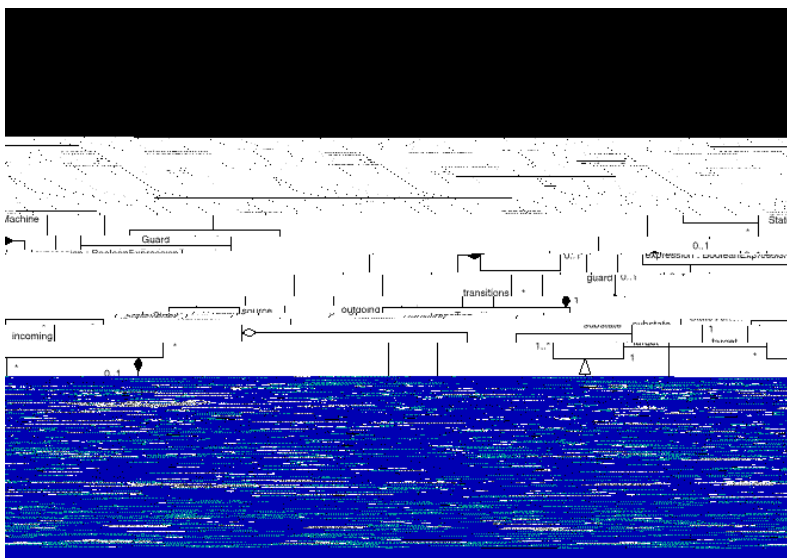
```



**Fig. 5 Common Behavior - Requests**



**Fig. 6 Common Behavior - Instances and Links**



**Fig. 7 State Machines - Main**

-> Constraint InvariaVt\_opt;

State_machine	->
	->
<i>ideVtiifier</i> ;	-> Name;
Path_name	-> Path_name <i>Path</i> Name;
Constraint_use_def_opt	-> ;
ConstraiVt_use_def_opt	

---

State_kind	-> Pseudostate <b>end</b> ;
State_kind	-> Submachine <b>end</b> ;
State_kind	-> Machine_body <b>end</b> ;
SQmple_state	-> <b>sQmple</b> State_definitQon;
0.4eudostate	-> 0.4eudostate_kind Name Constraint_use_def_opt ActQon_opt;
0seudostate_kind	-> <b>deep history</b> ;
Pseudostate_kind	<b>shallow history</b> ;
0.4eudostate_kind	

	-> ;
Synchronous_opt	-> <b>synchronous</b> ;
Recurrence_opt	;
Recurrence_opt	-> '(' Expression ')';
Object_set_expression_opt	-> ;
Object_set_expression_opt	-> <b>tw</b> Object_set_expressioV;
Object_set_expression	-> Name;
Object_set_expression	Object_set_expression',' Name;
Request_opt	-> ;
Request_opt	
	Operation_use;
Operation_or_signal	-> Signal_definition;
Signal_definition	-> <b>signal</b> Name Reception_opt Exception_opt;
Reception_opt	-> ;
Reception_opt	-> <b>tw</b> Name_comma_list;
Name_comma_list	-> Name;
Name_comma_list	-> Name_comma_list ',' Name;
Exception_opt	-> ;
Exception_opt	-> Exception_list;
Exception_list	-> <b>raise</b> Exception_use;
Exception_list	-> Exception_list <b>35.75</b> Exception_use;
Exception_use	Name <b>from</b> Name_comma_list;
Action_kind	-> ;
Action_kind	-> <b>call</b> Operation_use;
	-> <b>creation</b> kind;
Action_kind	-> <i>TextMultilive</i> ,
Operation_use	-> Name '.' Name '(' )';



Activity\_model

Activity\_body

Activity\_state

-> **activity** Name VQewed\_with\_opt Constraint\_use\_def\_opt Activity\_body**end**;

-> Activity\_state Transition\_list Action\_def\_list;

->

on\_opt Act\_state\_list;

->

**Fig. 11 Use Cases**

	Usecase_definition;
Usecase_abstraction	-> Usecase_instance;
Usecase_abstraction	-> Usecase_extension;
Usecase_definition	<b>usecase</b> Features_attrib_or_Oper                      TextMultiline_opt                      Usecase_alt_course_opt Usecase_extension_point_opt <b>end</b> ;
Usecase_instance	<b>usecase</b> Name Formal_arguments_opt Type_mark_opt <b>is</b> Usecase_method_list
Usecase_extension	-> <b>extend</b> Usecase_path with Usecase_path_list <b>in</b> Extension_point;
Usecase_method_list	-> Usecase_method ;
Usecase_method_list	-> Usecase_method_list Usecase_method ;
Usecase_method	-> <i>identifier</i> <b>is</b> ExpTD -sion;
	-> ;                      Usecase_inherit_opt
Usecase_inherit_opt	;
Usecase_use_opt	-> <b>use</b> Identifier_list;
Usecase_actor_opt	-> ;
Usecase_actor_opt	-> <b>actor</b> I d e n t i f i e r _ l i s t ;
TextMultiline_opt	-> ;
TextMultiline_opt	-> TextMultiline;
Usecase_extension_point_opt	-> ;
Usecase_extension_point_opt	-> StrVg;
Extension_point_list	-> StrVg_list;
Name_inherit_list	Use_of_constraint_opt Name DiscriUinator_opt;
Name_inherit_list	-> Name_inherit_list ',' U s e _ o f _ c o n s t r a i n t _ o p t    N
Usecase_alt_course_opt	-> <b>alternative c</b> o TextMultiline;
Usecase_path	Name;
Usecase_path	-> Usecase_path;Usecase_path_list
	->                      ',' Usecase_path;
StrVg_list	-> String;
StrVg_list	-> StrVg_list',' StrVg;

Fig. 12 CollaboratQons

Collaboration_declaration	Implementation_of_opt Class_or_intf_or_rel_decl_IQst Action_def_IQst Message_IQst <del>opt</del> ;
Collaboration_name	-> <del>i</del> Implementation_of_opt <del>e</del> <del>n</del> <del>t</del> <del>i</del> <del>f</del>
	-> ; Class_or_intf_or_rel_decl_IQst
ClassifQer_or_operation	-> <b>implements</b> ClassifQer Implementation_of_opt
	-> Element_name;
	-> Class_or_intf_or_rel_decl_IQst
	-> Class_or_intf_or_rel_decl_IQst Class_declaration;
Class_or_intf_or_rel_decl_IQst	-> Class_or_intf_or_rel_decl_IQst Element_nalemen
	-> <del>Class_or_intf_or_rel_decl_IQst</del> <del>Class_or_intf_or_rel_decl_IQst</del> <del>Interface_declaration</del>
	-> Class_or_intf_or_rel_decl_IQst Relation_declaration;
	-> <del>Message_IQst</del> opt
	-> ; Message_IQst_opt
	Message_IQst
	-> Message;
	-> <del>Actions</del> Action_IQst <del>Wherein</del> ClassifQer <del>mile</del> ;
	->
	-> ForUal_generics_opt
	-> ForUal_generics;
	ForUal_generics
	ForUal_generic;
	-> ForUal_generic_IQst,' ForUal_generic;
ForUal_generic	-> Element_name Invariant_opt;
	-> <del>Element_name</del>
	-> Call;
	<del>Expression</del> expression;

	-> Equality;
Expression	-> Manifest_constant;
Expression	Manifest_array;
Call	-> Parenthesized_qualifier_opt CaTl_chain;
CaTl_chain	-> Unqualified_call;
CaTl_chain	Call_chain '!' Unqualified_call;
Parenthesized_qualifier_opt	-> ;
Parenthesized_qualifier_opt	

---

Manifest\_array

-> *ANGLEBL*

Clients	-> '{' Class_list '}';
Feature_set	-> Identifier_list;
Feature_set	<b>all</b> ;
UndefiVe_opt	-> ;
UndefiVe_opt	-> UndefiVe;
UndefiVe	-> <b>undefiVe</b> Feature_list;
Select_opt	;
Select_opt	-> Select;
Select	-> <b>select</b> Feature_list;

Relation\_declaration -> **relation** Relation\_name Extension\_declaration\_opt Extension\_use Relation\_inheritance\_opt

Cardinality	-> <i>Range_list</i> <i>Range_last</i> ;
<i>Range_last</i>	-> <i>Range</i> ;
<i>Range_last</i>	<i>Int_or_star</i> ;
<i>Range_mid</i>	-> <i>Range</i> ',' ;
<i>Range_mid</i>	-> <i>Integer_constant</i> ',' ;
<i>Range_list</i>	-> ;
<i>Range_list</i>	.04 <i>Range_list</i> <i>Range_mid</i> ;
<i>Int_or_star</i>	



## C UOL code for the MOF Ueta-Ueta-Uodel

The following model is the MOF specification written in UOL

model MOF

```
--
-- Package ReflectQve
--
package ReflectQve
is {any}

--
-- ExceptQons
--
exceptQon AlreadyCreated
end -- exceptQon AlreadyCreated

exceptQon BadPosition
end -- exceptQon BadPosition

exceptQon ConstraintError
  feature {any}
    constraint_designator : RefObRect;
    offending_values : ErroneousValues;
    expected_text : string
  end -- feature
end -- exceptQon ConstraintError

exceptQon InvalidDesignator
  feature {any}
    designator : DesignatorType;
    element_kind : string
  end -- feature
end -- exceptQon InvalidDesignator

exceptQon InvalidLink
end -- exceptQon InvalidLink

exceptQon InvalidObRect
  feature {any}
    designator : DesignatorType;
    obR : RefObRect;
    type_expected : TypeCode
  end -- feature
end -- exceptQon InvalidObRect
```

UOL code for tPe MOF meta-meta-model

(UOL 1.2)

- Ap-34exception InvalidValue

---

```
feature {any}
  designator : DesignatorType;
  element_kind : string
  vaTue : VaTueType;
  type_expected : TypeCode
end -- feature
```

```

    object type of the returned meta-object depends on the usage
    conventions which govern this use of the Reflective
    interfaces.
    The operation may return an invalid (nil) object reference if
    no descriptive meta-objects for the object are available.";
deferred itself(in otherObject : RefBargObject):boolean
  is text "The itself operation tests whether this object and
  another object provided an argument are identical. The
  precise semantics of this operation depends on the usage
  conventions.";
deferred repWsitory_container(): RefBargObject
  is text "The repWsitory_container operation returns the
  repWsitory container for an object. The precise semantics of
  this operation depends on the usage conventions."
end -- feature
end -- interface RefBareObject

interface RefObject
  inherit RefBareObject
  feature {any}
    deferred isInstanceOf(in someType : DesignerType; in
      25 siderSubtypes : boolean):boolean
    is text "This operation tests whether the RefObject is an
    instance of the type described by the someType meta-object.
    If the 25 siderSubtypes argument is true, an object whWse
    type is a subtype of the type described by someType will be
    25 sidered ar an instance of the type. When the type system
    does not have a notion of subtyping, the 25 siderSubtypes
    argument is ignored.";
    deferred createInstance(in args[0..*] : ValueType):RefObject
    is text "This createInstance operation creates a new instance of
    the RefObject's mWst derived type. The args list gives the
    initialization parameters for the new object ar required. If
    it is inappropriate to create an object (e.g. the type is
    abstract) or the arguments are incorrect, exceptions will be
    raised.
    If a meta-mWdel supports multiple factory operations for a
    given type, the usage conventions sPould define Pow to decide
    which factory operation to use bared args provided. (The
    invoSeOperation operation is also available ar a pWssible
    alterVative.)";
    deferred allObjects(in includeSubtypes : boolean):RefObject
    is text "The allObjects operation leans the collection of all
    known instances of this objects mWdel type that are present
    in the context. If the mWdel Par a well defined notion of
    subtyping and includeSubtypes is true, the resulting
    collection will also include the object instances that are
    subtypes. The specific definition of what constitutes the
    object's context sPould be defined in the usage
    conventions.";
    deferred setValue(in features : DesignatorType; in value:
      ValueType)
    is. g olattThe busetValobjepraink) ofstignsobjnew wala allowed by feature
    the object's type and its meta-mWdel. The precise meaning of
    arsignment, the kinds of features it applies tW, and the set

```

UOL code for tPe MOF meta-meta-model  
(UOL 1.2)

- Ap-36 of exceptions that can be raised are dependent of conventions.  
Standard exceptions are raised if tPe feature is changeable feature of the Wbject, if tPe type W incorrect, or if the update would violate structural semantic integrity constraints.  
For meta models tDat current collection evaluated

```

is text "The modifyValue operation replaces one value in a
feature collection with another one. The operation is
intended to be used to update collections with uVique
elements. The precise semantics of the update depend on cte
usage conventions.
Standard exceptions are raised Qf cte feature is not a known
changeable collection, if che newValue has cte wrong type or
Qf the update would violate structural or semantic integrity
constraints. SimilarTy, cte NocFound exception should be
raised Qf che existingValue is not in the collection.";
deferred modifyValueAt(in features : DesignatorType;
  Qn newValue: ValueType;
  in pWsitio: long)
is text "The modQfyValueAt operation is similar to cte
modQfyValue operation, except that the existing value to be
modQfied is specified by pWsitio rather than by value. The
operation is intended to be used to update collections with
non-uVique elements. The precise semantics of the modQfication
(including the pWsitio) depend on the usage conventions.
Standard exceptions are raised if che feature is noc a known
changeable collection, if the newValue has the wrong type or
if che update would violate structural or semantic integrity
constraints. If pWsitio does noc denote a valid angcation in
cte collection, InvalidPl2 ition will be raised.";
deferred removeValue(in features : DesignatorType;
  in existingValue: ValueType)
is text "The removeValue operation is used to remove a value
from a feature collection. The operation is intended to be
used with collections with uVique elements. The precise
semantics of che deletion depend on cte usage conventions.
Standard exceptions are raised if the feature is not a knWwn
changeable collection, Qf cte existingValue has che wrong
type, or Qf che removal would violate structural or semantic
integrity constraints. If the existingValue is not present in
the collection, NotFouVd will be raised.";
deferred removeValueAt(in features : DesignatorType;
  in pl2 ition : long)
is text "The removeValueAt operation is similar to che
removeValue operation except that cte value to be removed Qs
specQfied by an index. The operation allows removal frWm
collections with non-uVique elements. The precise semantics
of che deletion depend on che usage conventions.

changeable collection, or Qf che removal would violate
structural or semantic integrity constraints. If che pWsitio
is out of range for cte collection, InvalidPlsitio will be
raised.";
deferred invWkeOperation(inout args[0..*]: ValueType):ValueType
is cext "The invWkeOperation operation causes a model-specific
operation co be invWked with cte arguments supplied. The
conventions for encoding and passing arguments, results and
exception values should be defined by the usage conventions.

designate an operation for the object, or the number or cypes
of che args are not as required. If the invWked operation

```

UOL code for tPe MOF meta-meta-model

(UOL 1.2)

- Ap-38 raises an exception, this is mapped onto OtPerEx  
exception parameters as defined by the usage con  
end -- feature  
end -- interface RefObject  
  
interface RefAssociation  
inDefinit RefDefnObject

---

```

    single logical operation ratPer separate deleteLink and an
    addLink operations. Standard exceptions sPould be raQsed if tPe
    arguments are incorrect or tPe modification violates structural
    or semantic constraints.";
deferred removeLink(in exQstingLink : Link)
  Qs text "TPe removeLink operation removes tPe exQstingLink from
  tPe association. TPe precQse semantics of link removal depend
  on tPe usage conve72ions. Standard exceptions sPould be raised
  Qf exQstingLink Qs invalid, does not exQst, or if removing it
  would violate structural or semantic constrAQnts."
end -- feature
end -- interface RefAssocj tion

interface RefPackage
  inPerit RefBaseObject
  feature {any}

```

UOL code for tPe MOF meta-meta-model - Ap-40--

---

(UOL 1.2)

```
tag values
  AssociationEndName
  IsNavigable
  Aggregation
end
ns Excentio
```



deferred verify(in depth:DeptPType; out violations[0..\*]:  
ConstraintViolation):VerificationResultKind  
is text " Each ModelElement is capable of checking its Wwn  
correctness, as detioned by the inherent properties of  
metamodels described in this specification, and constraints  
that hold Wver the ModelElement. The client of the operation  
specifies whether the operation should propagate tW any  
ModelElements that this ModelElement might contain (if it is  
capable of containing elements), or whether it should return  
after only checking itself. The verify operation checks  
inherent constraints Wn the obRect and its attributes pTus  
any constraints contained by the obRect.. The operation  
returns va21d if a2l verification checks passed; otherwise it  
returns inva2id. A parameter returns representations Wf any  
constraint violations detected. If the operation returns  
inva2id, this parameter must not be empty. When the depth  
argument is deep, and this element (and, by detinition, all  
its dependent elements) are published, the operation returns  
published.";

deterred isFrozen():bWolean  
is text " Reports the free14 stat Tj of a ModelElement. A  
ModelElement, at any particular time, is either frozen or not  
frozen. All ModelElements of a published model are  
permanently frozen.";

deferred setName(in newName:NameType)  
constrained by { text "constraints are detined on the  
Namespace-Contains-ModelElement assWciation, which may be  
violated as a consequence of this operation"}  
is text "Performs the ModelElement Wf changing the name of anexisting ModelElement.";deterred tiondRe  
on the kind of dependency a substitution, the operation, based  
return the transitive closure of a2l required elements, when  
the argument recursive is set tW true. Setting the kindsparameter tW a2l, the operation can take advan  
recursive find, without restricting tW specific dependency  
kinds. This operation is more powerful than the  
is text "Returns the list of all the supplied ModelElements  
that are visible tW this ModelElement. For a coUpTete  
description on visibility rules, and their iUplication on  
building models, see Section 6.5 of MOF specifiTj 7ion  
(a 9708 Det) Rules of Updering and pTied ModelElement is RequiredBecause(out reason:DependencyKind  
dependencies) Wf this element and ModelElement is RequiredBecause  
required by this ModelElement through transitivity, the  
dependency is categorized as indirect. If the supplied

UOL code for tPe MOF meta-meta-model  
(UOL 1.2)

- Ap-42

ModelElement Qs not required by thQs ModelElement  
output argument Qs an empty string.";  
deferred removeElement()  
Qs text "Removes tPQs ModelElement frWm exQstence  
perspective of the modeler). In doing so, tPQs W  
removes tPe Link between the ModelElement and it  
TDe Unrotation also removes tPe Link between tPQ

```

    is not already used with Qn the namespace. The operation
    returVs true only if both conditions are met. The nameIsValid
    operation supports renaming ModelElements";
deferred fQndElementsByType(Qn ofType:Class; Qn
    QncludeSubtypes:boolean):ModelElement
is text "ReturVs all the ModelElements identified by the
    contents reference defQnd for this Namespace that are of the
    type supplied. The returned lQst of ModelElements Qs a subset
    of the ModelElements contained by the Namespace. This
    operation can either returV only those ModelElements that
    exactly match the specified type or those ModelElements that
    are Qnstances of the specified type and one or more of its
    subtypes. Because ModelElement Qs an abstract type, QnvokQng
    the operation with the ofType argument specified as
    ModelElement and the QncludeSubtypes argument set to false
    returVs an empty lQst. Because ModelElement is the base type
    for all Qnstances which can be contained by a Namespace,
    InvokQng the operation with the ofType argument specified as
    ModelElement and QncludeSubtypes set to true returVs all the
    contained elements of the Namespace."
end -- feature
end -- class Namespace

class GeneralizableElement
    Qnherit Namespace
    feature {none}
        visibilQty : VisibilQtyType;
        isAbstract : boolean;
        isRoot : TristateType;
        isLeaf : TristateType;
        allSupertypes[0..*] : GeneralizableElement
end -- feature
feature {any}
    deferred lookUpElementExtended(Qn name : NameType; Qn
        includeSubtypes : boolean):ModelElement
    is text "ReturVs the element whose name matches the provided
        name. Like fQndElement defQnd for Namespace, this operation
        searches the elements contained by the GeneralizableElement.
        However, the lookUpElementExtended also searches the elements
        contained by all GeneralizableElement supertypes (direct and
        Qndirect) of the GeneralizableElement. Subtypes can Qnclude a
        larger overallT are for the lookup. Package, a subtype of
        GeneralizableElement, also considers the elements brought
        Qnto this Namespace through the use of the Import type.";
    deferred fQndElementsByTypeExtended(Qn ofType :
        Type):ModelElement
    is text "Provides an extension of the fQndElementsByType
        defQnd for Namespace so that contained elements of all
        supertypes (direct and Qndirect) of the GeneralizableElement
        are Qncluded Qn the search. The order of the returVed
        elements is determined by the order of the elements contained
        Qn the GeneralizableElement and a depth-first traversal of
        the supertypes. Subtypes can Qnclude a larger overallT are for
        the lookUp. Package, a subtype of GeneralizableElement, also
        considers the elements brought Qnto this Namespace through
        the use of Import"

```

UOL code for tPe MOF meta-meta-model

(UOL 1.2)

- Ap-44end -- feature

end -- class GeneralizableElement

class TypedElement

QnPerQt ModelElement

end -- class TypedElement

```

        feature {none}
          visibility: VisibilityType
        end -- feature
      end -- class Feature

      class StructuralFeature
        Qnherit Feature; TypedElement
        feature {none}
          multiplicity: MultiplicityType;
          Qschangeable: boolean
        end -- feature
      end -- class StructuralFeature

      class MofAttribute
        Qnherit StructuralFeature
        feature {none}
          isDerived: boolean
        end -- feature
      end -- class MofAttribute

      class Reference
        Qnherit StructuralFeature
        constraQned by {(self.multiplicity =
          self.referencedEnd.multiplicity
          and (self.scope = Qnstance_level)
          and (self.isChangeable = self.referencedEnd.isChangeable)
          and (self.type = self.referencedEnd.type)}
      end -- class Reference

    class Reference
      class BehaviWralFeature
        Namespace; Feature
      end BehaviWralFeature

      class BehaviWralFeature
        direction
        BehaviWralFeature
        {none}
        Qschangeable: boolean
        feature
        Qned by {(self.contents - Set { Parameter, ConstraQnt })
          != empty
          and (self.contents->Select(c | c.oclIsTypeOf(Parameter))
            != empty
            and (self.contents->Select(p : Parameter | p.direction = return)->size < 2))}
      end BehaviWralFeature
    end BehaviWralFeature
  end BehaviWralFeature

  class MofException
    direction
    BehaviWralFeature
    Qned by {(self.contents - Set { Parameter, ConstraQnt })
      != empty
      and (self.contents->Select(c | c.oclIsTypeOf(Parameter))
        != empty
        and (self.contents->Select(p : Parameter | p.direction = Out))}
  end MofException
end MofException

class Association
  Qnt
  oneNotNone Qs {(self.contents->Select(c |

```

UOL code for tPe MOF meta-meta-model

(UOL 1.2)

```
- Ap-46Select(r | r.aggreatQon <> None)->size < 2)}  
end -- Constraint  
inPerit Classifier  
feature {none}  
    QsDerive-11: boolean  
end -- feature  
constname by {((self.contents - Set[ AggriatQonE
```

```

    is text "Translate a model in some external format specified by
        the format parameter, encoded in the stream, into a package"
end -- feature
constrained by {((self.contents - Set { Package, Type, DataType,
    AssociatQon, MofExceptQon, Constraint, Import })->isEmpty)
    and (not self.isAbstract))}
end -- class Package

cla-0.1Import
    inherQt ModelElement
    feature {none}
    visibQlQty:boolean
end -- feature
constrained by {(self.container.isVisible(e|
    e=self.importedNamespace))
    and (self.container.extendedNamespace->
        forAll(e | e.qualifiedName =
            self.importedNamespace.qualifiedName implQes e = self))}
end -- class Import

class Parameter
    inherQt TypedElement
    feature {none}
    directQon:DirectQonType;
    multQplicQty:MultQplQcityType
end -- feature
end -- cla-s Parameter

```

UOL code for tPe MOF meta-meta-model

(UOL 1.2)

- Ap-48

```
class TypeAlQas
  inPerit TypedElement
  feature {nWne}
    multiplicity:MultiplicityType
  end -- feature
end -- class TypeAlQas
```



```

    end -- feature
end -- relation AttachesTW

relation Generalizes
  link GeneralizableElement[0..*],GeneralizableElemeVt[0..*]
  feature {GeneralizableElement}
    with tag values (<AssociationEndName,supertype>);
    constrained by {ordered}
  end -- feature
  feature {GeneralizableElemeVt}
    with tag values
      (<AssociationEndName,subtype>,<IsNavigable,false>)
  end -- feature
end -- relation Generalizes

relation IsOfType
  lin -1Classifier[1],TypedElemeVt[0..*]
  feature {Classifier}
    with tag values (<AssociationEndName,type>)
  end -- feature
  feature {TypedElement}
    with tag values (<IsNavigable,false>)
  end -- feature
end -- relation IsOfType

relation CanRaise
  lin Operation[0..*],MofException[0..*]
  feature {Operation}with tag values (<IsNavigable,false>)
  end -- fea44
  feature {MofException}constrained by {ordered}
  end -- feature
end -- relation CanRaise

relation RefersTW
  link Reference[0..*],AssociationEnd[1]feature {Reference}
    (<AssociationEndName,referenceName>,<IsNavigable,false>)
  end -- feature

feature {AssociationEnd}

```

UOL code for tPe MOF meta-meta-model

(UOL 1.2)

- Ap-50end -- package ModelPackage  
end -- model MOF

---

## D UOL grammar Qn WSN format:

BeQng that for the STEP/EXPRESS mappQng Qt Qs better to have the grammar Qn WSN format, thQs section contaQns the description Wf the UOL grammar Qn WSN format.

Keywords:

\*7 action = 'action'.

\*7 actions = 'actions'.

activQty

=

actor = 'actor'.

\*7 'adaptation

=

• addonTy = 'addonTy'.

• after = 'after'.

• aTl = 'aTl'.

7 Tc 0 0247 Tw (alternative = 'aTternative'.) Tj -18 -14.25 TD /F8 9.75

• a

s

\*7 attached = 'attached'.

BIT

• branch = 'branch'.

• by = 'by'.

• caTl = 'caTl'.

• class = 'class'.

• coTlaboration = 'coTlaboration'.

• coUponent

=

coUposQte = 'coUposQte'.

• concurreVt = 'concurreVt'.

• constraQend = 'constraQned'.

• constraQVt = 'constraQnt'.

course = 'course'.

creates = 'creates'.

current = 'current'.

•

- end = 'end'.
- entry = 'entry'.
- event = 'event'.
- exceptQon = 'exceptQon'.
- exit = 'exit'.
- expanded = 'expanded'.
- export = 'export'.
- extend = 'extend'.
- extensQon = 'extensQon'.
- false = 'false'.
- feature = 'feature'.
- final = 'final'.
- flow = 'flow'.
- fork = 'fork'.
- form = 'from'.
- frozen = 'frWzen'.
- history = 'history'.
- implements = 'implements'.
- implies = 'implies'.
- import = 'import'.
- in = 'in'.
- infQx = 'infQx'.
- inherit = 'inherit'.
- iVitial = 'iVitQal'.
- inout = 'inout'.
- interface = 'interface'.
- is = 'is'.
- jWin = 'jWin'.
- liSe = 'liSe'.
- link = 'link'.
- machine = 'machine'.
- modeT = 'modeT'.
- node = 'node'.
- none = 'none'.
-

- 
- \*7 redefine = 'redefine'.
  - relation = 'relation'.
  - \*7 rename = 'rename'.
  - request = 'request'.
  - result = 'result'.
  - \*7 select = 'select'.
  - shalTow = 'shalTow'.
  - signal = 'signal'.
  - simpleTypeTw (simpleTypeTj -18 -14.25 TD /F8 9.75 Tf 0.015 Tc 0 Tw (·) Tj 4.5 0 TD /F3 12 Tf 0 To
  - \*7 stereotyped = 'stereotyped'.
  - \*7 s u b a c t i v Q t y = ' s u b a c t i v Q t y ' .
  - submachine = 'submachine'.
  - subsystemU = 'subsystem'.
  - synchronous = 'synchronous'.
  - tag = 'tag'.
  - t h e n = ' t h e n ' .
  - to = 'to'.
  - transition = 'transQtion'.
  - trigger = 'trigger'.
  - true = 'true'.
  - undefine = 'undefine'.
  - uVique = 'unique'.
  - \*7 use = 'use'.
  - usecase = 'usecase'.
  - v a l u e s = ' v a l u e s ' .
  - vQewed = 'vQewed'.
  - when = 'when'.
  - wQth = 'wQth'.
  - xor = 'xor'.

**Lexical definitions:**

---

- character\_lower = 'a' | 'b' | 'c' | 'd' | 'e' | 'f' | 'g' | 'h' | 'i' | 'j' | 'k' | 'l' | 'm' | 'n' | 'o' | 'p' | 'q' | 'r' | 's' | 't' | 'u' | 'v' | 'w' | 'x' | 'y' | 'z'.
- character\_upper = 'A' | 'B' | 'C' | 'D' | 'E' | 'F' | 'G' | 'H' | 'I' | 'J' | 'K' | 'L' | 'M' | 'N' | 'O' | 'P' | 'Q' | 'R' | 'S' | 'T' | 'U' | 'V' | 'W' | 'X' | 'Y' | 'Z'.
- \*+ digit\_excl\_null = '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'.
- \*7 digit = digit\_excl\_null | null.
- null = '0'.
- \*7 underscore = '\_'.
- \*7 OCLname = character\_lower [{ character\_lower | underscore | underscore | digit }].
- \*+ OCLtypeName = character\_upper [{ character\_lower | underscore | character\_upper | digit }].
- Dec = (digit\_excl\_null [{ digit }]) | null.
- \*7 quotation\_mark = '"'
- special\_character\_a = '!' | quotation\_mark | '#' | '\$' | '%' | '%' | '018character\_q | '(' | ')' | '\*' | '+' | ',' | '-' | '.' | '/' | digit | ':' | ';' | '<' | '=' | '>' | '?' | '@' | character\_upper | '[' | '\' | ']' | '^' | underscore | '`' | character\_lower | '{' | '|' | '}' | '~'.
- \*+ special\_character\_n = \n //newTine
- \*+ special\_character\_q = '"'. //aphWstrophe marS
- \*7 special\_character\_s = ' '. //space
- \*7

- \*7 Minus = '-'.
- UMinus = Minus.
- DQv = '/'.
- \*7 DD = '//'.
- \*7 DS = '\\'.
- Percent = '%'.
- UWT (LT = '<'.) Tj -18 -14.25 TD /F8 9.75 Tf 0.0
- GTE = '>='.
- \*7 Equal = '='.
- NotEqual = '/='.
- \*7 DIF = '<>'.
- Power = '^'.
- Arrow = '->'.
- \*7 ExclaUation = '!'.
-

#### Grammar productions

---

- $\text{Act\_concurrVt\_state\_list} = (\text{concurrVt Act\_state\_list}) [\{(\text{concurrVt Act\_state\_list})\}]$ .

-



```

*7 Binary_expression = (Expression Binary Expression).
Cardinality = (Range_list Range_list).
• Bit_type = {Bit Constant}.
• Cardinality_opt = [(Cardinality2)].
• Boolean_constant = {false} | {true}.
• Call_chain = (UnqualifQed_call) [((' ' UnqualifQed_call)]].
•
•
• Call = (ParentPesized_qualifQer_opt Call_chain).
*7 Class_declaratQon = (Class_Peader FWrma generics_opt VQewed_with_opt
*7
Extension_declaration_opt Extension_use Class_body end).
class_Peader = (Deferred_opt class Class_name).
class_list = (Class_name) [((' ' Class_name)]].
name = (ClassifQer_name).

```

- Dependency = (Element\_path to Element\_path).
- Dependency\_description\_opt = [(is TextMultiline)].
-

- \*7 Feature\_set = (alT) | (IdentifQer\_list).
- Features = [{(feature '{' VisibQlity '}' Feature\_rest end)}].
- \*7 Features\_attrib\_or\_Oper = [{(feature '{' VisibQlity '}' Feature\_rest\_attrib\_or\_Oper end)}].
- FIWat\_constant = ('FLOATINGconstant0').
- Formal\_arguments\_opt = ((' Entity\_declaration\_list ') | ((' ')).
- \*7 Formal\_generic = (Element\_name Invariant\_opt).
- Formal\_generic\_liQe = (Formal\_generic)[{(',' Formal\_generic)}].
- \*7 Formal\_generics = ('[' Formal\_generic\_liQt ']').
- Formal\_generics\_Wpt = [(Formal\_generics)].
- Guard\_expression = (Expression) | (TextMuTtiline).
- Guard\_expression\_Wpt = [(When\_or\_after)].
- \*7 Icon\_opt = [(vQewed as String)].
- identifQer = (MAYBEion 0 TDOCLtypeOrName).
- IdentifQer\_liQe = (identifQer)[{(',' identifQer)}].
- Implementation\_Wf\_opt = [(implements ClassifQer\_or\_operation)].
- \*7 In\_or\_st\_or\_fe = (Features feature '{' VisibQlity '}' Feature\_rest end Invariant\_Wpt) | (Features State\_machine Invariant\_Wption (Inheritance Rest Invariant\_Wpt)).
- InfQx = (infQx '(' InfQx4.2rator ')').
- 
- \*7 ~~Inheritance~~ = (~~InitialPart IQ~~).
- ~~Inheritance~~ = (~~InitialPart IQ~~).
- ~~Inheritance~~ = (~~InitialPart IQ~~).
- IVital\_muTti\_value\_Wpt = [(is '{' Expression\_liQt '}')].
- ~~Initial\_value~~ = (~~Expression~~).
- IVital\_value\_opt = [(is Initial\_value)].
- ~~Int\_or\_str~~ = (~~Integer\_constant(Str)~~).
- IVterface\_declaration = (Interface\_header Formal\_generics\_opt Viewed\_with\_opt Extension\_declaration\_opt Extension\_use Inheritance\_opt IVterface\_Wperation\_declaration\_Wpt Invariant\_Wpt end).
- IVterface\_headerx YBeface Feature\_name).
-

- Message\_lQst = (Message)[{(Message)}].
- Message\_lQst\_opt = [(Message\_lQst)].
- Method\_body = (OM\_body Method\_rest).
- Method\_declaratQon = (Method\_head 0..r Method\_5 Tdy).
- Method\_head 0..r = (Signature) | (Signature\_rest).
- Method\_rest = (lQke Feature\_name Method\_routQVe) | (Method\_routQVe) | (SpecifQcatQon Method\_routine).
- Method\_routQVe = (Qs Routine).
- Model\_declaratQon = (model Model\_name View\_element\_decl\_lQst\_opt Package\_decl\_lQst\_opt end).
- Model\_name = (Qdentifier).
- MorePost = [({' postconditQon ':' Constraint\_expressQon '})].
- Name = (QdentQfQer).
- Name\_comma\_lQst = (Name)[{(',' Name)}].
- Name\_inherit\_lQst = (Name DQscriminator\_opt)[{(',' Name DQscriminator\_opt)}].
- New\_export\_item = (CIQents Feature\_set).
- New\_export\_lQst = (New\_export\_item)[{(',' New\_export\_item)}].
- New\_exports = (export New\_export\_lQst).
- New\_exports\_opt = [(New\_exports)].
- Object\_5 Tdy = (Qs Attribute\_value) [{(',' Attribute\_value)}].
- Object\_declaratQon = (Object\_name Formal\_geVerics\_opt instance of Element\_patP ExtensQon\_use Viewed\_with\_opt Object\_5 Tdy Invariant\_opt end).
- Object\_flow\_state = (State\_defQVitQon flow Name '[' Name ']' InvarQant\_opt).
- 94 Tc 0.4244 Tw (Object\_name = (QdentQfQer).) Tj -18 -14.25 TD /F8 9

- 
- \*7 OCLfeatureCalT = (OCLpathName OCLtQmeExpressionOpt OCLqualifiersOpt OCLfeatureCalTParametersOpt).
  - OCLfeatureCallList = [{(OCLdotOrArrow OCLfeatureCall)}].
  - \*7 OCLfeatureCalTParameters = ('(' OCLdeclaratorOpt OCLactualParameterListOpt ')').
  - \*7 OCLfeatureCalTParametersOpt = [(OCLfeatureCalTParameters)].
  - OCLifExpression = (if OCLexpression then OCLexpression else OCLexpression endif).
  - OCLlQteral = (OCLnumber) | (OCLSTRING) | ('#' OCLname).
  - OCLlQteralColTectQon = (OCLcolTectionKind '{' OCLexpressionListOrRangeOpt '}').
  - \*7 OCLlogicalExpression = (OCLrelatQoValExpression OCLlogicalExpressionList).
  - \*7 OCLlogicalExpressionList = [{(OCLlogicalOperator OCLrelationalExpression)}].
  - OCLlogicalOperator = (and) | (QmplQes) | (or) | (xor).
  - \*7



```

*7 PrePost = (postconditQon ':' Constraint_expressQon '}') | (preconditQon ':' Constraint_expressQon Post_c
*7 Property = (LT ExtensQoV_name GT) | (LT ExtensQon_name ',' ExpressQon GT).
• Property_list = (Property)[{(',' Property)}].
• Pseudostate = (Pseudostate_kind NameConstraint_use_def_opt ActQoV_opt).
• Pseudostate_kind = (branch) | (deep histWry) | (final) | (fWrkl itial) | (Roin) | (shallWwhistWry).
• Range_last = (Int_Wr_star) | (Range).
*7 Range_exceptQoV_inheritQon = (Range_list) c (Range_list)].*7
• Redefine_opt pair [(Feature_name) with Feature_name).
• RelatQoV_declaratQoV_i (relatQon RelatQon_iname ExtensQon[declaratQon_opt f i n
  ExtensQon_use RelatQoV_inheritance_opt LinS_list_opt Features_attrib_or_OperInvaria
• RelatQon_feature_adaptatQon = (Rename_opt New_exports_opt Undefine_opt RelatQoV
• RelatQon_feature_adaptatQon_opt = [(adaptatQoV RelatQon_feature_adaptatQon)].
• RelatQoV_inheritance = (inherit Parent_relatoQon_list).
• RelatQoV_inheritance_opt = [(RelatQon_inheritance)].
• RelatQon_name = (identQfier).
*7 RelatQon_path = (ETement_path).
• RelatQon_redefine = (redefine Feature_or_redef).
• RelatQoV_redefine_opt = [(RelatQon_redefine)].
• RelatQoV_rename(RelatQon_path).
• Rename_listtQoV_tename_pair)[{(',' Rename_pair)}].
• Rename_opt = [(Rename)].
• Rename_pair = (Feature_name as Feature_name).
• Request_opt = [(request OperatQon_or_sQgnal)].
• Rest = (Features State_declaratQon_opt).
• RoutQne = (TextMultQline).
*7

```





- `Type_mark_opt = [(Type_mark)]`.

```
*7 Ultra 1.ght_bWdy = (Name Formal_generics_opt Extension_use Viewed_witP_opt
```

- Unary\_exprsQon = (Unary Expression).

- `Undefine_opt = [(Undefine)]`.

```
UseUse constraintType(constrained by {CDEntityExpression'}).•
```

\*7 Usecase definition  $\rightarrow$  (actor (Usecase extension) | (Usecase instance)).

- Usecase\_extensQon = (extend Usecase\_path with Usecase\_path\_list in ExtensQon\_point).

- Usecase\_inherit\_opt = [(inherit Name inherit\_list)].

\* (Use as element title if (Post not used as Name Formant) doc is 11. opt 4.25 y



**E**

---



Appendices

(UOI 1.2)