International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Z.119

(02/2007)

SERIES Z: LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

Formal description techniques (FDT) – Application of formal description techniques

---

Guidelines for UML profile design

ITU-T Recommendation Z.119

ITU-T Z-SERIES RECOMMENDATIONS

LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

FORMAL DESCRIPTION TECHNIQUES (FDT)	
Specification and Description Language (SDL)	Z.100–Z.109
Application of formal description techniques	**Z.110–Z.119**
Message Sequence Chart (MSC)	Z.120–Z.129
Extended Object Definition Language (eODL)	Z.130–Z.139
Testing and Test Control Notation (TTCN)	Z.140–Z.149
User Requirements Notation (URN)	Z.150–Z.159
PROGRAMMING LANGUAGES	
CHILL: The ITU-T high level language	Z.200–Z.209
MAN-MACHINE LANGUAGE	
General principles	Z.300–Z.309
Basic syntax and dialogue procedures	Z.310–Z.319
Extended MML for visual display terminals	Z.320–Z.329
Specification of the man-machine interface	Z.330–Z.349
Data-oriented human-machine interfaces	Z.350–Z.359
Human-machine interfaces for the management of telecommunications networks	Z.360–Z.379
QUALITY	
Quality of telecommunication software	Z.400–Z.409
Quality aspects of protocol-related Recommendations	Z.450–Z.459
METHODS	
Methods for validation and testing	Z.500–Z.519
MIDDLEWARE	
Processing environment architectures	Z.600–Z.609

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation Z.119

Guidelines for UML profile design

Summary

ITU-T provides several languages for telecommunication applications (see Recommendations in the X.680 series, Z.100 series, Z.120 series, Z.130 series, Z.140 series and Z.150 series). Although each of these languages can be used on its own, they can also be used as a family of system design languages by showing the relationship between the languages. For this purpose, UML defined by the object management group (OMG) can be used as a framework to bind the languages together and capture the relationships through the creation of a UML profile for each language. The purpose of this Recommendation is to provide a guideline on how to read and write such profiles.

Source

ITU-T Recommendation Z.119 was approved on 13 February 2007 by ITU-T Study Group 17 (2005-2008) under the ITU-T Recommendation A.8 procedure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

CONTENTS

ITU-T Recommendation Z.119

Guidelines for UML profile design

1 Scope

This Recommendation is intended to provide a common and pragmatic approach to writing, reading and editing an ITU-T Recommendation that is a UML profile for a language within the family of ITU-T system design languages (see Recommendations in the X.680 series, Z.100 series, Z.120 series, Z.130 series, Z.140 series and Z.150 series). Each of UML profile Recommendation for such an ITU-T system design language (hereinafter referred to as an 'ITU-T language') provides a basis for using UML models with tools for the ITU-T language, adapting UML tools to support the ITU-T language, and being able to integrate models in and tools for the ITU-T language with other models and tools that fit into the UML framework. Such profiles are therefore useful to both toolmakers and language users, though the benefit for language users is most likely to be through supporting tools.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[OMG UML] OMG Unified Modeling Language: Superstructure version 2.1.1 formal/ 2007-02-05

3 Definitions

This Recommendation defines the following terms:

3.1 ITU-T language: A formal language defined in an ITU-T Recommendation that is used as a system design language such as the languages defined by Recommendations in the X.680 series, Z.100 series, Z.120 series, Z.130 series, Z.140 series and Z.150 series.

3.2 second level clause: The term second level clause shall be used to two-digit numbered text passage.

3.3 system design language: A formal notation that has a well-defined syntax and semantics that is used for some aspect of the engineering systems ranging from statements of requirements for procurement or in standards to the deployment of implemented systems or networks.

3.4 third level clause: The term third level clause shall be used to three-digit numbered text passage.

3.5 top level clause: The term top level clause shall be used to denote single-digit numbered text passage.

4 Abbreviations

This Recommendation uses the following abbreviations:

BNF	Backus-Naur Form of syntax description
UML	Unified Modeling Language 2.0 (see [OMG UML])
UML-SS	OMG UML-2.0 Superstructure Specification (see [OMG UML])

5 Profile basics

Although UML provides a substantial framework for the description of models, the UML-SS leaves many aspects undefined (or open to semantic variation) including some of the concrete syntax of the language. The result is that for many kinds of model the well-defined parts of UML are inadequate to define a useful language, and a profile is needed. Most UML tools implicitly provide such profiles to make UML practical, for example binding the action language to C++ or Java. For an ITU-T language, a specific profile is provided binding UML to the semantics of the ITU-T language as defined by the ITU-T Recommendation(s) for that language and therefore closing semantic variation points.

The framework of UML covers many different aspects of engineering from requirements capture through to configuration and deployment. By contrast, each ITU-T language focuses on more specific modelling such as user requirements expression, definition of protocol data types, modelling message sequences, specification and description of a system of state machines, definition of tests, or interface and deployment characteristics. Therefore, a UML profile for an ITU-T language is applicable only to those parts of UML covering the same domain: UML is constrained by the profile.

The purpose of a UML profile is essentially the following:

– to define how to associate extra information with model elements in a UML model;

– to define additional concepts that are not defined by UML, but that can be based on the existing UML concepts;

– to constrain UML so that the model can be mapped to the ITU-T language.

From a pragmatic point of view, a profile is simply a UML package stereotyped with keyword "profile". The "profile" package contains stereotypes that define:

– tagged values (the attributes of the stereotype);

– what model element to extend (the 'extends' relation to a metaclass).

A simple example is shown in Figure 1.

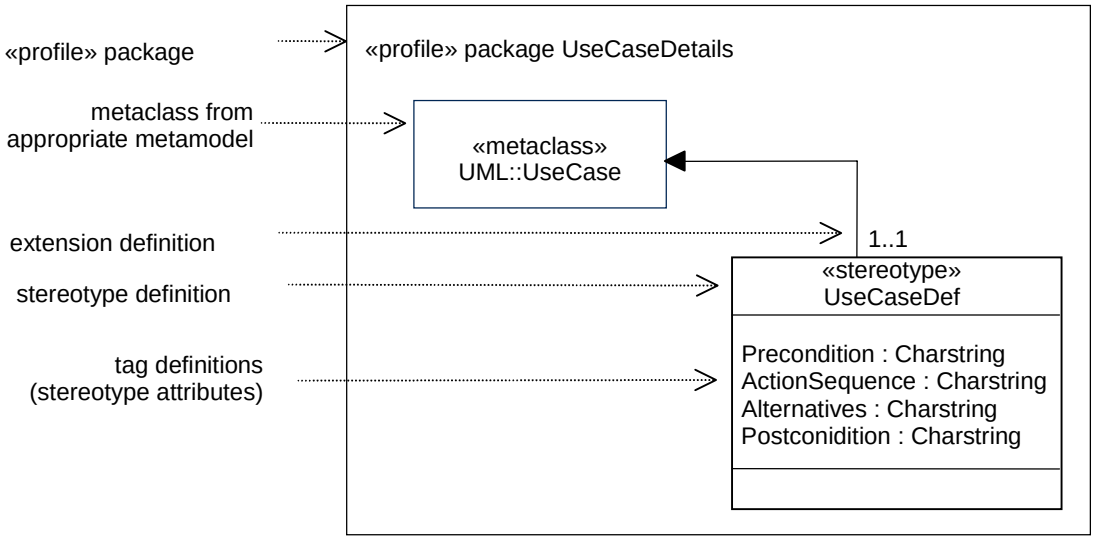

Figure 1

In this example, we have defined a stereotype called UseCaseDef that extends the built-in UML concept UseCase. The purpose of this is to add some extra items to all use cases (the concept defined by UseCase). In this case, the extras are Precondition, ActionSequence, Alternatives and Postcondition attributes of the stereotype.

The stereotype extends the metaclass with multiplicity [1..1]. This is essential because it will enforce that from a user's point of view whenever a use case is created in a package where the profile is applied, the use case will have the specified properties. Moreover, if the stereotype has the same name as the metaclass (in this case UseCase instead of UseCaseDef), in application models the name UseCase always refers to the extended definition.

This is the mechanism that should be used in ITU-T language profiles whenever there is a need to add extra information to the built-in UML concepts. In the text of a profile, multiplicity should be specified as "multiplicity [1..1]" or "multiplicity of [1..1]" to emphasize that there is exactly one. The form "multiplicity [1]" should never be used because this has the same appearance as a reference. There is a derived boolean attribute of the association between metaclass and the stereotype called isRequired that can be shown as "{required}" on the association at the same end as the multiplicity [1..1]. The "{required}" notation is redundant compared with the notation for multiplicity of [1..1], but if these diagrams are drawn it is preferable to use the multiplicity only because isRequired is derived from the multiplicity and there is no explicit notation for "not required".

A second use of a profile is to introduce new concepts. A simple example is shown in Figure 2.

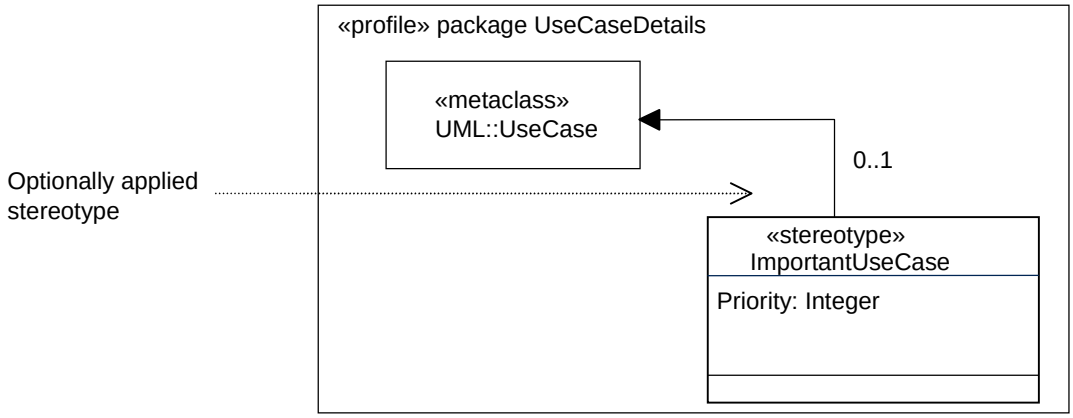

Figure 2

In this example, we have introduced a new concept called "ImportantUseCase". From a user's point of view, it would be visible as a stereotype that optionally can be applied to use cases. The key point to notice is that the multiplicity of the extends relation is "0..1". This means that some UseCases will be ImportantUseCases, but not all of them.

This is the mechanism that should be used in ITU-T language profiles, when there is a need to add concepts that have no direct mapping to a built-in UML concept.

Occasionally the same UML concept has two (or more) alternative extensions, one of which should be applied. In this case, the original UML concept should be extended with the stereotype multiplicity [1..1] with the same name as the original concept, and this stereotype is extended with the alternative extensions with multiplicity [0..1] with different names. The stereotype of the original UML concept shall have an additional constraint that every extension shall have the constraints and properties of one (and only one) of the alternative extensions.

In UML, a profile is *applied* to a package using a dashed arrow notation with the keyword 'apply' within quotes as in Figure 3.

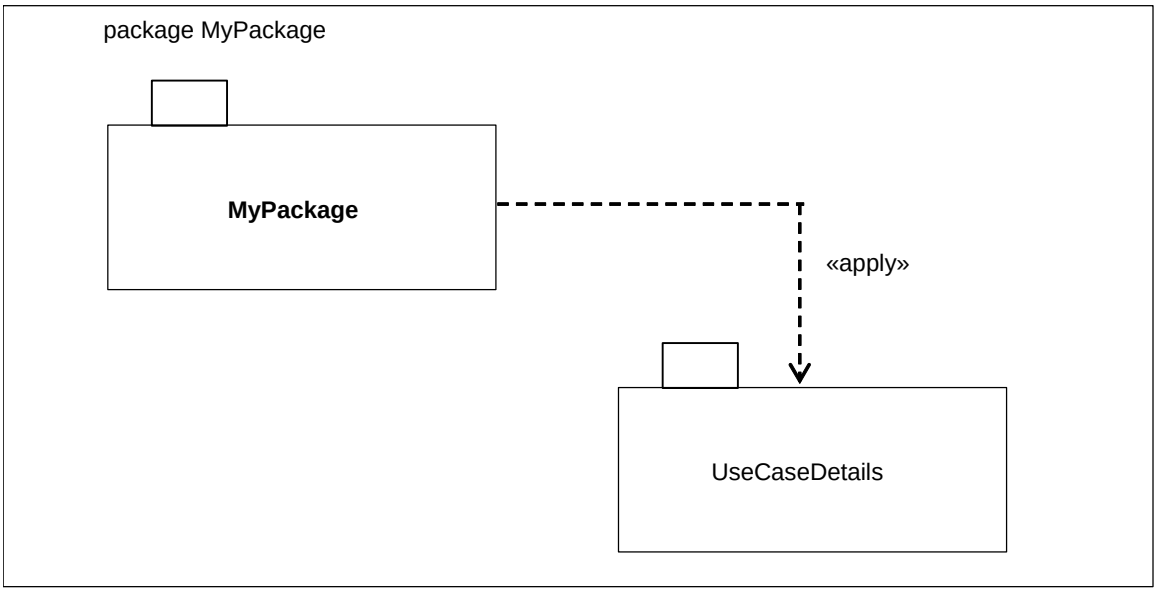

Figure 3

However, it should be noted that UML tools also usually provide simpler ways to apply profiles, so from a user's point of view, the relevant ITU-T profile would simply be selected in the user interface of the tool and would then immediately give access to the definitions in the profile.

To summarize, the main points of the UML profile definitions that should be used when defining ITU-T language profiles:

– stereotypes that extend the built-in UML metaclasses with multiplicity [1..1] are used to add extra information to built-in UML concepts;

– stereotypes that extend the built-in UML metaclasses with multiplicity [0..1] are used to define new concepts based on the built-in UML concepts.

6 A profile document

Table 1 – Arrangement of elements (typical)

Element	Clause numbers
Introductory material[a)]	
Title page[b)], Summary, Source[b)], Keywords (optional), Foreword[b)], Table of contents[b)], Introduction (describes the purpose of the profile)	None
Recommendation core material	
Title	None
Scope (describes the scope of the profile, and the following second level clauses)	1
Conformance (stating rules for conformance)	1.1
Notation (stating notation guidelines for the UML)	1.2
Restrictions to languages (for example 'Restriction to MSC and UML', stating the parts of the ITU-T language and UML-SS covered)	1.3
References (which shall include the relevant ITU-T language Recs, UML-SS, OCL and (if applicable) MOF)	2
Definitions (shall state 'the terms and definitions given in ITU-T Recs' for the ITU-T Language 'and the following applies and any term defined below applies if it is also defined elsewhere' followed by a list of definitions)	3
Abbreviations (which shall include OCL, UML, and UML-SS)	4
Conventions (documentation of conventions used in the profile, which shall refer to **Common Conventions** defined in this Recommendation so that different profiles are consistent). This clause may also contain other common material that applies to the whole profile, in which case 5.1 should be 'Conventions'.	5 (or 5.1 see description)
Text of Recommendation defining the profile	6 onwards
Annexes (an integral part of the Rec., may contain formal)	A onwards
Appendix I (summary of relevant parts of the UML metamodel)	I onwards [a)]
Appendix II onwards (not an integral part of the Rec.)	II onwards [a)]
Electronic attachment	– [c)]
Bibliography and Index (optional)	None

[a)] These elements are not considered as part of the Recommendation proper.

[b)] These elements are provided by TSB.

[c)] Electronic attachment may correspond to a clause, an annex, an appendix or be a non-numbered item.

Table 1 elaborates the 'Author's guide for drafting ITU-T Recommendations'. Specific guidelines for the structure and content of a profile Recommendation are further defined below.

6.1 Common conventions

The **Conventions** clause of the Recommendation defining the profile, should define all the meta-languages and notations used in the profile, and may define other items (such as name resolution, drawing style, formal notations for mapping, translation and transformation) used throughout the profile if they are not defined in an additional clause. As far as possible, reference should be made to other Recommendations (such as ITU-T Rec. Z.111 for notations to define grammars) or other standards, which shall be included in the **References** clause of the Recommendation defining the profile.

It is possible to include the following conventions by reference to this clause of this Recommendation, but it may be preferable to include the text below noting it is the same as this clause of this Recommendation:

– A term in the profile Recommendation is a sequence of printing characters usually being either an English word or a concatenation of English words that indicate the meaning of the term.

– An underlined term refers to a UML term or a term defined by a stereotype of the profile Recommendation. A term starting with an uppercase letter by convention is the name of a metaclass.

– A term preceded by the word "stereotype" names a UML stereotype used for the profile Recommendation, according to the stereotype concept defined in the UML Superstructure specification documentation (usually in a phrase "The stereotype X extends the metaclass X" where X is a term) and the term shall start with a capital letter. If the multiplicity of the stereotype is [1..1], the stereotype is required (that is the derived attribute isRequired of the Extension association between the extended metaclass and the stereotype is true). If the multiplicity of the stereotype is [0..1], the stereotype is not required.

– A term enclosed in << and >> as brackets refers to a stereotype described by the profile Recommendation (the term is not underlined).

– A term in Courier font refers to some text that appears in the model either as written by a user or to represent some text created from the expansion of a shorthand notation for the relevant construct.

– The profile includes only model elements of the UML-SS that are specifically mentioned in the profile Recommendation, or that are a superclass (recursively) of a model element included in the profile.

– There is an error in the profile Recommendation if there is any inconsistency between the natural language and any corresponding formal notation definition in the profile Recommendation.

The **Conventions** clause is allowed to contain an extensive description of a common item. In this case, it is suggested to rename the clause and have second level clauses. For example, if the clause defines name resolution as well as other conventions, the top level clause would be named "**5 Conventions and name resolution**" with second level clauses "**5.1 Conventions**" and "**5.2 Name resolution**".

6.2 Text of Recommendation defining the profile

The structure of a profile document should contain the following parts:

– **Stereotype summary**: Clause 6, giving an overview of the stereotypes defined in the profile. Preferably this should be one or more UML diagrams giving the stereotype definitions but it can also simply be a list or table of the stereotypes that are defined.

– **Stereotypes**: Typically clauses 7 onwards are used to group the stereotypes into relevant parts, with one second level clause for each stereotype needed in the profile. One suggested way to structure is to have top level clause for each package in the UML metamodel that is used, and include the stereotype chapters as second level clauses. There may also be a need for additional clauses to describe issues not specifically related to one package or stereotype.

– **UML metamodel**: This should be an appendix to the profile document that summarizes the relevant parts of the UML metamodel. This is not strictly necessary, but may make the profile document easier to understand.

The main part of a profile document is the definition of the stereotypes themselves. Each stereotype second level clause should be structured as described in clause 6.3.

6.3 Structure of stereotype second level clauses

The text at the start of the second level clause should define the stereotype itself. An informal introduction to the purpose and semantics of the stereotype of the concept should be given in a note. For example:

The stereotype PassiveClass extends the merged metaclass Class with multiplicity [0..1].

NOTE 1 – It is intended to capture the semantics of passive classes corresponding to object data types in SDL.

or in the (exceptional) case that an informative note is not needed:

The stereotype Signal extends the metaclass Signal with multiplicity [1..1].

A stereotype shall have multiplicity [1..1] or multiplicity [0..1]. The name of a stereotype shall start with a capital letter. If the multiplicity is [1..1], the name of the stereotype should normally be the same as the name of the UML concept it extends.

After this description, third level clauses Attributes, Constraints, Semantics, Notation and References should be given. It is suggested that if there is nothing specific for a third level clause, the clause is not omitted but the phrase given below for that third level clause is used, so that the numbering of third level clauses is consistent. However, if the majority of third level clauses just repeat the phrases given, the profile will be much longer than is necessary, so in this case, omitting the third level clause may be preferable so that the whole document is shorter. In that case, this strategy should be documented in the conventions.

Underlining is used when referring to UML metamodel terms and italics (or another distinct presentation style appropriate to the ITU-T Language) should be used when referring to ITU-T language semantic entities. Except where it is defined (as above) a stereotype should be referred to (without underlining) within << and >> and usually precedes the base UML metaclass. For example:

<<PassiveClass>> Class

NOTE 2 – For a simple profile, the stereotypes could be top level clauses and Attributes, etc. second level clauses.

6.3.1 Attributes

This third level clause defines the attributes for extra information that is needed to capture the ITU-T language concept.

If no extra information is needed, third level clause should contain the text 'No additional attributes'.

There shall be a list of attributes definitions separated with dashes optionally with a heading "Stereotype attributes". Before or after the list additional, normal paragraphs are allowed to provide additional description. Continuation lines and paragraphs of list items are indented to distinguish them from normal paragraphs.

Each item in the attribute list shall be a single list item, preferably in a single paragraph. The start of each paragraph formally defines the attribute with a term for the attribute and the kind of the attribute. The term names the attribute and shall start with a lower case letter and is separated from the kind by a colon. The kind is the term for an element in the UML-SS model or the profile followed by an optional multiplicity. The notation used is the normal notation for attributes in the UML model, and in this context the attribute name and kind of the attribute are not underlined.

The remainder of the list item paragraph (and any additional paragraphs of the same list item) is optional text in natural language that gives further explanation and description of the attribute. In this text and any other natural language text, defined attribute names or other terms from UML-SS and profile are underlined.

6.3.2 Constraints

This third level clause defines the constraints that are needed for the stereotype to make sure that the UML model will be well formed with respect to the ITU-T language concept it is intended to capture. The constraints should be expressed in natural language and where possible also in OCL (object constraint language, the formal constraint notation that is part of UML). There is an error in the profile Recommendation if there is any inconsistency between the natural language and OCL constraints.

6.3.3 Semantics

This third level clause defines the mapping between the UML metaclass and stereotype and the corresponding ITU-T language element. As an example, for SDL this is a mapping to the SDL abstract grammar. See clause 7, *Specifying semantics*, below for more details.

6.3.4 Notation

This third level clause specifies the notation for instances of stereotyped model elements. See clause 8, *Specifying notation*, below for more details.

6.3.5 References

This third level clause specifies where to find the related sections of the UML-SS and the ITU-T Recommendation(s). While strictly this is not needed, it is invaluable for readers so that the relevant parts of UML-SS and ITU-T Recommendation(s) can be found and the effect of the profile understood. The references to the UML-SS should show the clause number in UML-SS, the name of the item and in brackets the UML packages that define the item. For example:

UML-SS 7.3.36 Operation (from Kernel, Interfaces)

7 Specifying semantics

The semantics clause shall contain a description of how:

– a stereotyped instance is mapped to the ITU-T language;

– each property of the extended UML metaclass and of the stereotype itself is mapped to ITU-T language elements.

The definition of the mapping from a stereotyped instance to an ITU-T language concept is done using a natural language statement as in the following example:

> a <<Package>> Package is mapped to a *Package-definition*.

The mapping of each attribute of the metaclass and the stereotype itself should be done with a natural language statement as in the following style:

> the name of the Package maps to the *Package-name*.

Additionally mapping using a formal mapping notation is permitted, but no specific formalism is suggested and it is left open whether the formal mapping should be placed in the Semantics third level clause or within an annex. There is an error in the profile Recommendation if there is any inconsistency between the natural language and formal mapping.

The mapping of each attribute of the extended metaclass, each superclass of this metaclass and of the stereotype itself shall be defined. The definition of these details is a major part of the profile document.

8 Specifying notation

In most cases, there is no need to specify the notation to be used for a concept because UML already contains a sufficient notation. In these cases, a simple statement like the following should be used:

> UML standard syntax is used.

When there is a need to introduce a new graphical symbol, the style used in the UML specification to introduce symbols should be followed. This essentially is composed of a textual description of the symbol followed by one or more examples of what the symbol might look like.

An example:

> an actor is represented by "stick man" icon with the name of the actor in the vicinity (usually above or below) the icon.

When needed, the text in the symbols should be defined using the BNF based concrete grammar rules specified in ITU-T Rec. Z.111. As far as possible, constraints on the language notation should be expressed in terms of the profile and UML metamodel elements, and placed in the **Constraints** third level clauses. Constraints relevant to notation only (such as the placement of text in relation to symbols, or the placement or need for separators) can be described with the notation.

The translation from the notation to the metamodel elements of the profile (or UML-SS) should be defined in the Notation third level clause, or for translation rules that apply generally (such as name resolution, significance of font styles) should be defined in the **Conventions** clause. The translation

shall be expressed in natural language, but it is allowed for the translation to also be expressed in a formal translation notation.

In some cases, notation is a short hand or model that is expanded by a transformation into a longer more canonical form of notation, possibly with new implicitly named elements being introduced. As far as possible, these transformations should avoid reference to elements to the ITU-T language, and preferably should avoid reference to profile and UML metamodel elements. The preferred style for transformations is to be expressed only in terms of notation elements.

In some cases only, a textual notation is defined for a concept. In particular, this is likely to apply for the syntax of actions or expressions, and grammar rules should be used to define the syntax.

9 An example second level clause

The following example is derived from a UML profile for SDL. For the purpose of the example some changes have been made, and the example is not intended to be the same as the Recommended UML profile for SDL.

9.1 Property

The stereotype Property extends the metaclass Property with multiplicity [1..1].

NOTE – It is intended to capture the semantics of attributes that correspond to variables and agent instance sets in SDL.

9.1.1 Attributes

Stereotype attributes:

– initialNumber: Natural [0..1] defines the initial number of instances created when an instance of the containing classifier is created.

9.1.2 Constraints

- A <<Property>> Property shall not have aggregation == shared.
- If a <<Property>> Property has aggregation == composite, the type shall be a <<PassiveClass>> Class with at least one structuralFeature or an <<ActiveClass>> Class.
- The initialNumber is included only if the <<Property>> Property has aggregation == composite and has a type that is an <<ActiveClass>> Class. The value of the InitialNumber shall not be less than the lowerValue or greater than the upperValue.
- isDerived shall be false.
- isDerivedUnion shall be false.
- if isReadOnly is true the type of the property shall be a DataType.
- if isReadOnly is true the defaultValue shall be a constant Expression.
- if a <<Property>> Property that has a type that is an <<ActiveClass>> Class and has aggregation == none, isReadOnly shall be false and the defaultValue shall be empty.

9.1.3 Semantics

A <<Property>> Property that has isReadOnly == false and has aggregation == none and has a type that is a <<PassiveClass>> Class or an Interface or a DataType (which includes PrimitiveType and Enumeration) is mapped to a *Variable-definition*. The name maps to the *Variable-name*. The type property maps to the *Sort-reference-identifier* that represents the type in the SDL abstract syntax. If the upperValue has a value larger than 1, the *Sort-reference-identifier* should reference the SDL predefined String datatype instantiated with the sort given by the type property as argument. The defaultValue defines the *Constant-expression*.

If the Property has a type that is a <<PassiveClass>> Class and the aggregation is composite the *SortReference* is defined using the *Value* modifier to create an anonymous value type based on the object type given by the type property.

A <<Property>> Property that has isReadOnly == true and has a type that is a <<PassiveClass>> Class or an Interface or a DataType (which includes PrimitiveType and Enumeration) is a read only variable in the UML model. When the UML is mapped to the SDL abstract syntax, the defaultValue defines a *Constant-expression* that is used to replace each reference to the read only variable. The <<Property>> Property therefore has the same characteristics as an SDL synonym.

A <<Property>> Property that has a type that is an <<ActiveClass>> Class and has aggregation == none is mapped to a *Variable-definition* of a pid sort. The Name defines the *Variable-name.* The type property defines the *Sort-reference-identifier* that represents the type in the SDL abstract syntax. If the upperValue has a value larger than 1, the *Sort-reference-identifier* should reference the SDL predefined String datatype instantiated with the pid sort given by the type property as argument.

A <<Property>> Property that has a type that is an <<ActiveClass>> Class and has aggregation == composite is mapped to an *Agent-definition*. The Name defines the *Agent-name.* The type property defines the *Agent-type-identifier* that represents the type in the SDL abstract syntax. The initialNumber defines the *Initial-number*. The upperValue defines the *Maximum-number*. If the initialNumber is omitted, the lowerValue defines the *Initial-number*. If both the initialNumber and lowerValue are omitted, the *Initial-number* is 1.

9.1.4 Notation

UML standard syntax <property> is used with the following extension and constraint. The initialNumber is optionally specified by as a slash followed by an <integer name> after the multiplicity. The property type <prop-type> shall not be omitted. That is:

<property> ::=
[<visibility>] **[/]**
<name> **:** <prop-type> [<open square bracket> <multiplicity> <close square bracket>]
[**/** <integer name>]
[**=** <default>]
[<left curly bracket> <prop-property > { **,** <prop-property> }* <right curly bracket>]

9.1.5 References

SDL: 9 Agents

12.3.1 Variable definition

UML-SS: 7.3.44 Property (from Kernel, Association Classes)

7.3.49 StructuralFeature (from Kernel)

7.3.52 TypedElement (from Kernel)

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	**Languages and general software aspects for telecommunication systems**

Printed in Switzerland
Geneva, 2007