INTERNATIONAL STANDARD

ISO 20022-2

Third edition
2013-05-01

Financial services — Universal financial industry message scheme —

Part 2: UML profile

Services financiers — Schéma universel de messages pour l'industrie financière —

Partie 2: Profil UML

Reference number
ISO 20022-2:2013(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 20022-2 was prepared by Technical Committee ISO/TC 68, *Financial services*.

This third edition cancels and replaces the second edition (ISO 20022-2:2007) which has been technically revised.

ISO 20022 consists of the following parts, under the general title *Financial services — Universal financial industry message scheme*:

— *Part 1: Metamodel*

— *Part 2: UML profile*

— *Part 3: Modelling*

— *Part 4: XML Schema generation*

— *Part 5: Reverse engineering*

— *Part 6: Message transport characteristics*

— *Part 7: Registration*

— *Part 8: ASN.1 generation*

ISO 20022-1:2013, ISO 20022-2:2013, ISO 20022-3:2013, ISO 20022-4:2013, ISO 20022-5:2013, ISO 20022-6:2013, ISO 20022-7:2013 and ISO 20022-8:2013 will be implemented by the Registration Authority by no later than the end of May 2013, at which time support for the concepts set out within them will be effective. Users and potential users of the ISO 20022 series are encouraged to familiarize themselves with the 2013 editions as soon as possible, in order to understand their impact and take advantage of their content as soon as they are implemented by the Registration Authority. For further guidance, please contact the Registration Authority.

For the purposes of research on financial industry message standards, users are encouraged to share their views on ISO 20022:2013 and their priorities for changes to future editions of the document. Click on the link below to take part in the online survey:

http://www.surveymonkey.com/s/20022_2013

Introduction

This International Standard defines a scalable, methodical process to ensure consistent descriptions of messages throughout the financial services industry.

The purpose of this International Standard is to describe precisely and completely the externally observable aspects of financial services messaging in a way that can be verified independently against operational messaging.

The trigger for the creation of this International Standard was the rapid growth in the scale and sophistication of messaging within financial services during the 1990s using ISO 15022. The financial services industry (from here on referred to as "the industry") created the first version of this International Standard as the successor to ISO 15022 in response to that trigger. Since ISO 15022, the industry has broadened the scope from securities to the entire industry for this International Standard.

This International Standard is based on open technology standards, which historically have evolved more rapidly than the industry itself. Consequently, this International Standard adopted a model-driven approach where the model of the industry's messaging can evolve separately from the evolution of the messaging technology standards. The period during which this International Standard has emerged followed the widespread adoption of the World Wide Web (the Web) for business. XML (eXtensible Mark-up Language) emerged as the *de facto* standard for document representation on the Web and it became the first syntax for ISO 20022.

The modelling process is further refined into three levels which, in addition to the messaging technology standard, is why this International Standard is based on four levels: the Scope level, the Conceptual level, the Logical level and the Physical level.

This four-level approach is based on the first four levels of the Zachman Framework. The remaining two levels of the Zachman Framework are equivalent to the implementations and the operational levels, respectively.

In ISO 20022-1, the first, second and third levels are described in UML (Unified Modelling Language) because it is widely supported and supports multiple levels of abstraction. The models created in accordance with this International Standard are technology independent in that they do not require any particular physical expression or implementation. Such models aim to describe all parts of the message exchange. The models form the definition of the protocol between participants exchanging messages. This International Standard defines a method that describes a process by which these models can be created and maintained by the modellers.

The models and the Physical level artefacts are stored in a central repository, serviced by a Registration Authority. This International Standard's repository is available on the World Wide Web and offers public access for browsing.

The Repository is organized into two areas:

— A DataDictionary containing the industry model elements likely to have further or repeated use.

— A BusinessProcessCatalogue that contains models describing specific message definitions and business processes, and physical syntax implementations.

This International Standard is organized into the following parts.

— ISO 20022-1 describes in MOF (Meta-Object Facility) the metamodel of all the models and the Repository.

— This part of ISO 20022 covers the UML profile, a grounding of general UML into a specific subset defined for this International Standard (to be used when UML is selected to define the models).

— ISO 20022-3 describes a modelling method to produce models for this International Standard.

— ISO 20022-4 covers XML schema generation rules to transform a Logical level model into a Physical level description in the syntaxes.

— ISO 20022-5 covers logical model alignment and reverse engineering of existing message syntaxes.

— ISO 20022-6 covers message transport characteristics that define the quality of service required by the business process definitions so that they can operate successfully.

— ISO 20022-7 describes the process of managing the registration of models and physical syntax implementations.

— ISO 20022-8 gives ASN.1 syntax generation rules to transform a Logical level model into a Physical level description in ASN.1.

Financial services — Universal financial industry message scheme —

Part 2: UML profile

1 Scope

This part of ISO 20022 defines the UML Profile for this International Standard. In essence, it defines how to use UML to create models that conform to the ISO 20022 Metamodel, which is defined in ISO 20022-1. In so doing, it defines a UML-based concrete syntax for the Metamodel. It does not preclude the specification of additional concrete syntaxes for the Metamodel, such as a textual concrete syntax.

The Profile defines how to represent in UML each of the Metamodel's Scope Level Elements (Level 1), Business Level Elements (Level 2) and Message Level Elements (Level 3), as well as Metamodel Elements that are scoped across the levels.

Therefore, the Profile covers all of the Metamodel's Packages, except for the following:

- ISO20022::Metamodel::ConceptualLevel::MessageTransport
- ISO20022::Metamodel::LogicalLevel::Reversing
- ISO20022::Metamodel::LogicalToPhysicalTransformation
- ISO20022::Metamodel::PhysicalLevel

The Profile also covers the ISO20022::TypeLibrary Package, upon which the Metamodel has some dependencies.

This part of ISO 20022 is only applicable when UML is used.

2 Normative references

ISO 20022-1, *Financial services — Universal financial industry message scheme — Part 1: Metamodel*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 20022-1 and the following apply.

3.1
Metamodel[1)]
the ISO 20022 metamodel

1) The reason that the defined term is "the metamodel" rather than "metamodel" is that sometimes the term "UML Metamodel" is used, and it would be incorrect in such cases to substitute "ISO 20022 metamodel" for "metamodel," since the result of the substitution would be "UML ISO 20022 metamodel."

3.2
Profile
the UML profile for ISO 20022

3.3
UML metamodel
OMG's metamodel of UML

4 How the Profile is specified

4.1 General

This clause explains the technique used to specify the Profile.

4.2 Package structure of the Profile

The internal Package structure of the Profile is similar to the internal Package structure of the Metamodel. Whereas the Metamodel's top-level Package is ISO20022::Metamodel, the Profile's top-level Package is ISO20022::Profile. The following are the top-level Packages within the ISO20022::Profile Package:

1) ISO20022::Profile::ScopeLevel;

2) ISO20022::Profile::DataTypes;

3) ISO20022::Profile::ConceptualLevel;

4) ISO20022::Profile::LogicalLevel;

5) ISO20022::Profile::ConceptualToLogicalTransformation.

The ISO20022::Profile::Conceptual Package contains two Packages:

ISO20022::Profile::Conceptual::Dynamic;

ISO20022::Profile::Conceptual::Static.

The ISO20022::Profile Package imports two Packages:

ISO20022::TypeLibrary::Enumerations;

ISO20022::TypeLibrary::XMLSchema.

NOTE The ISO20022::Profile Package contains no Package that corresponds to the ISO20022::Metamodel::ScopeToConceptual Package, because it is not necessary for the Profile to define any Stereotypes in order to implement that Metamodel Package.

4.3 Basic organization of the Profile specification

Clause 5 systematically outlines the Metamodel, Package by Package, defining how each Metamodel element maps to UML (it does not include the Packages of the Metamodel that the Profile does not cover, which are listed in the Scope). Within each Metamodel Package, the Metaclasses are treated in alphabetical order, and

then any DataTypes contained in the Package are treated in alphabetical order[2]. For each Metaclass and DataType, the Corresponding UML Element is specified.

After outlining the ISO20022::Metamodel Package and its sub-Packages, Clause 5 covers the ISO20022::TypeLibrary Package and its sub-Packages, defining how each element maps to UML.

In many cases, the definition of how an element maps to UML refers to Stereotypes and Tag Definitions that are normatively defined in Annex A.

4.4 Properties of the Metamodel and UML — Tag Definitions

In outlining the Metamodel, Metaclass by Metaclass, this part of ISO 20022 lists each of the Properties of the Metaclass along with the defined approach for modelling that Property in UML. In some cases, the definition states that the Property in question maps to a Tag Definition defined in Annex A.

4.5 Properties of the Metamodel that correspond to existing UML Properties

In many cases, a Property of a Metaclass from the Metamodel corresponds to a UML Property already defined in the OMG's UML specification. This part of ISO 20022 explicitly defines these correspondences. In such a case, there is no need to define a Tag Definition to realize the Property in UML.

In a number of these cases, the Profile defines that the corresponding UML Property is the non-navigable memberEnd of an Association in the UML specification, meaning that the Property cannot be captured in an ISO 20022-compliant UML model. In essence, this means that the Metamodel supports the back-pointer while UML does not, and there is no compelling reason to define a Tag Definition to fill the gap. The burden of tracking and setting the back-pointer via a Tagged Value would fall entirely on the modeller, unless additional code with specific knowledge of this back-pointer had been added to the UML tool.

4.6 AssociationEnds

UML 1.x has a Metaclass named AssociationEnd. UML 2.x does not have a Metaclass of that name; the Metaclass has been replaced by the "memberEnd" Property of Association. Consequently, this part of ISO 20022 refers frequently to a "memberEnd" of an Association, whereas UML 1.x discourse would refer to an "AssociationEnd." For readability, the quotation marks have been omitted when referring to this Property, even though all other Property names are in quotation marks when they appear in narrative text.

4.7 Constraints

Profile Constraints are expressed in OCL. The Constraints are motivated by the fact that the Profile is an implementation of the ISO 20022 Metamodel, and they restrict the freedom of the UML modeller. UML models that do not adhere to the Constraints are not ISO 20022-compliant.

4.8 Figures

The definitions in Annex A of the Profile Stereotypes and Tagged Definitions include figures that use UML Profile notation to graphically depict the Stereotypes and Tag Definitions. The figures also depict the UML Metaclasses that the Stereotypes extend, and the Stereotypes' superclasses where applicable. As is typical in specifications of MOF Metamodels and UML Profiles, the diagrams duplicate some of the information contained in the narrative text. The narrative text is normative and the diagrams are non-normative.

4.9 How modellers choose which UML Diagrams to create

A complete overview of when to use which UML diagram can be found in ISO 20022-3.

2) The DataTypes that lie within the Metamodel Package and its contained Packages are DataTypes used to specify the Metamodel. On the other hand, the DataTypes that lie within the ISO20022::TypeLibrary Package and its contained Packages are DataTypes used by modellers to specify ISO 20022–compliant financial models.

5 Mapping the Metamodel's elements to UML

5.1 General

This clause defines how each element of the Metamodel is realized in UML.

5.2 UML realization of ISO20022::Metamodel

5.2.1 General

This subclause defines how the elements of the ISO20022::Metamodel Package map to UML.

5.2.2 Metaclass: BusinessProcessCatalogue

— Corresponding UML Element: A <<BusinessProcessCatalogue>>-stereotyped Package.

— Mappings of the Metaclass's Properties:

repository: not implemented in the Profile[3)];

topLevelCatalogueEntry: corresponds to UML Package's "packagedElement" Property, which points to <<TopLevelCatalogueEntry>>-stereotyped UML Elements contained by the <<BusinessProcessCatalogue>>-stereotyped Package.

5.2.3 Metaclass: CodeSet

— Corresponding UML Element: A <<CodeSet>>-stereotyped Enumeration.

— Mappings of the Metaclass's Properties:

identificationScheme: captured via <<CodeSet>>'s "identificationScheme" Tag Definition;

code: corresponds to UML Enumeration's "ownedLiteral" Property, which points to <<Code>>-stereotyped EnumerationLiterals;

derivation: corresponds to the "supplierDependency" memberEnd of the Association between UML's NamedElement and Dependency Metaclasses (the other memberEnd of the Association is named "supplier"). The <<CodeSet>>-stereotyped Enumeration is the NamedElement playing the "supplier" role and a <<Trace>>-stereotyped Dependency plays the "supplierDependency" role (another <<CodeSet>>-stereotyped Enumeration is the client end of the Dependency);

trace: corresponds to the "clientDependency" memberEnd of the Association between UML's NamedElement and Dependency Metaclasses (the other memberEnd of the Association is named "client"). The <<CodeSet>>-stereotyped Enumeration is the NamedElement playing the "client" role and a <<Trace>>-stereotyped Dependency plays the "clientDependency" role (another <<CodeSet>>-stereotyped Enumeration is the supplier end of the Dependency);

length (inherited from Text): captured via <<CodeSet>>'s length Tag Definition;

minLength (inherited from Text): captured via <<CodeSet>>'s minLength Tag Definition;

maxLength (inherited from Text): captured via <<CodeSet>>'s maxLength Tag Definition.

3) It is not realized in the Profile because there is no explicit element in the Profile that corresponds to the Metamodel's Repository Metaclass (see 5.2.8).

5.2.4 Metaclass: Code

— Corresponding UML Element: a <<Code>>-stereotyped EnumerationLiteral.

— Mappings of the Metaclass's Properties:

Code value: corresponds to the "body" Property of the OpaqueExpression that plays the "specification" role for the UML EnumerationLiteral.

5.2.5 Metaclass: Constraint

— Corresponding UML Element: a <<Constraint>>-stereotyped Constraint.

— Mappings of the Metaclass's Properties:

— expression: corresponds to the "body" Property of the OpaqueExpression that plays the "specification" role for the UML Constraint;

— expressionLanguage: corresponds to the "language" Property of the OpaqueExpression that plays the "specification" role for the UML Constraint.

5.2.6 Metaclass: DataDictionary

— Corresponding UML Element: a <<DataDictionary>>-stereotyped Package.

— Mappings of the Metaclass's Properties:

— repository: not implemented in the Profile[4];

— topLevelDictionaryEntry: corresponds to UML Package's "packagedElement" Property, which points to <<TopLevelDictionaryEntry>>-stereotyped UML Elements contained by the <<DataDictionary>>-stereotyped Package.

5.2.7 Metaclass: IdentifierSet

— Stereotype: an <<IdentifierSet>>-stereotyped Class.

— Mappings of the Metaclass's Properties:

identificationScheme: captured via <<IdentifierSet>>'s "identificationScheme" Tag Definition;

length (inherited from Text): captured via <<IdentifierSet>>'s length Tag Definition;

minLength (inherited from Text): captured via <<IdentifierSet>>'s minLength Tag Definition;

maxLength (inherited from Text): captured via <<IdentifierSet>>'s maxLength Tag Definition.

5.2.8 Metaclass: Repository

— Corresponding UML Element: there is no corresponding UML element. ISO 20022-compliant models do not contain an explicit element corresponding to the Metamodel's Repository Metaclass. The overall ISO 20022 Repository implicitly realizes this Metaclass.

— Mappings of the Metaclass's Properties: N/A

4) It is not realized in the Profile because there is no explicit element in the Profile that corresponds to the Metamodel's Repository Metaclass (see 5.2.8).

5.2.9 Metaclass: RepositoryConcept

— Corresponding UML Element: a NamedElement, stereotyped by one of the concrete descendants of the <<RepositoryConcept>> Stereotype.

— Mappings of the Metaclass's Properties:

name: corresponds to NamedElement's "name" Property;

definition: corresponds to the "ownedComment" Property of UML Element. NamedElement is a descendant of Element and thus inherits the "ownedComment" Property. "ownedComment" is of type Comment, and the "definition" populates the "body" Property of Comment;

example: captured via <<RepositoryConcept>>'s "example" Tag Definition;

registrationStatus: captured via <<RepositoryConcept>>'s "registrationStatus" Tag Definition;

removalDate: captured via <<RepositoryConcept>>'s "removalDate" Tag Definition;

semanticMarkup: captured via <<RepositoryConcept>>'s "semanticMarkup" Tag Definition;

constraint: corresponds to <<Constraint>>-stereotyped UML Constraints. For each such UML Constraint, the value of the "constrainedElement" Property points to the RepositoryConcept that owns the Constraint.

5.2.10 Metaclass: TopLevelCatalogueEntry

— Corresponding UML Element: a NamedElement, stereotyped by one of the concrete descendants of the <<TopLevelCatalogueEntry>> Stereotype.

— Mappings of the Metaclass's Properties:

  — businessProcessCatalogue: corresponds to the "owner" Property, which UML NamedElements inherit from Element, and which points to a <<BusinessProcessCatalogue>>-stereotyped Package.

5.2.11 Metaclass: TopLevelDictionaryEntry

— Corresponding UML Element: a NamedElement, stereotyped by one of the concrete descendants of the <<TopLevelDictionaryEntry>> Stereotype.

— Mappings of the Metaclass's Properties:

  — dataDictionary: corresponds to the "owner" Property, which UML NamedElements inherit from Element, and which points to a <<DataDictionary>>-stereotyped Package.

5.2.12 Metaclass: Trace

— Corresponding UML Element: a <<Trace>>-stereotyped Dependency.

— Mappings of the Metaclass's Properties: Trace has no Properties of its own.

5.2.13 DataType: Cardinality

— Corresponding UML Element: MultiplicityElement.

— Mappings of the Metaclass's Properties:

- isOrdered: corresponds to the “isOrdered” Property of MultiplicityElement;
- isUnique: corresponds to the “isUnique” Property of MultiplicityElement;
- maximumOccurrence: corresponds to the “upperValue” Property of MultiplicityElement. If maximumOccurence contains 'UNBOUNDED' then the "upperValue" is literal UnlimitedNatural with value "infinity";
- minimumOccurrence: corresponds to the “lowerValue” Property of MultiplicityElement.

5.3 UML realization of ISO20022::Metamodel::ScopeLevel

5.3.1 General

This subclause defines how the elements of the ISO20022::Metamodel::ScopeLevel Package map to UML.

5.3.2 Metaclass: BusinessProcess

- Corresponding UML Element: a <<BusinessProcess>>-stereotyped UseCase.
- Mappings of the Metaclass’s Properties:
  - businessRole: corresponds to the Actor memberEnd of an Association between a <<BusinessRole>>-stereotyped Actor and a <<BusinessProcess>>-stereotyped UseCase;
  - included: corresponds to the “includingCase” Property of the UML Include Metaclass, which is accessed via UseCase’s “include” Property;
  - includer: corresponds to the non-navigable memberEnd of the Association between UML’s Include and UseCase Metaclasses, where the navigable memberEnd is “includingCase”;
  - extender: corresponds to the non-navigable memberEnd of the Association between UML’s Extend and UseCase Metaclasses, where the navigable memberEnd is “extendedCase”;
  - extended: corresponds to the “extendedCase” Property of the UML Extend Metaclass, which is accessed via UseCase’s “extend” Property;
  - businessProcessTrace: corresponds to the “ownedBehavior” Property that UseCase inherits from BehavioredClassifier, pointing to the <<BusinessTransaction>>-stereotyped Interactions that the <<BusinessProcess>>-stereotyped UseCase owns.[5)]

5.3.3 Metaclass: BusinessRole

- Corresponding UML Element: a <<BusinessRole>>-stereotyped Actor.
- Mappings of the Metaclass’s Properties:
  - businessProcess: corresponds to the UseCase memberEnd of an Association between a <<BusinessRole>>-stereotyped Actor and a <<BusinessProcess>>-stereotyped UseCase;
  - businessRoleTrace: corresponds to the use of an instance of a <<BusinessRole>>-stereotyped Actor to represent a Lifeline[6)] in a <<BusinessTransaction>>-stereotyped Interaction.[7)]

5) In the Metamodel, this relationship is captured via the BusinessProcessTrace Metaclass, which relates BusinessProcesses to BusinessTransactions.

5.4 UML realization of ISO20022::Metamodel::ScopeToConceptualTransformation

5.4.1 General

This subclause defines how the elements of the ISO20022::Metamodel::ScopeToConceptualTransformation Package map to UML.

NOTE The Profile implements all of the Properties of this Package's Metaclasses via existing UML elements, so Annex A defines no Stereotypes corresponding to the elements of this Package.

5.4.2 Metaclass: BusinessProcessTrace

— Corresponding UML Element: a link between a <<BusinessProcess>>-stereotyped UseCase and a <<BusinessTransaction>>-stereotyped Interaction. Such a link is an instance of the UML Metamodel's Association between BehavioredClassifier and Behavior.[8)]

— Mappings of the Metaclass's Properties:

  — businessProcess: corresponds to the "context" property that UML Interaction inherits from Behavior, which points a <<BusinessTransaction>>-stereotyped Interaction to a <<BusinessProcess>>-stereotyped UseCase.[9)]

  — businessTransaction: corresponds to the "ownedBehavior" Property that UseCase inherits from BehavioredClassifier, which points a <<BusinessProcess>>-stereotyped UseCase to <<BusinessTransaction>>-stereotyped Interactions.

5.4.3 Metaclass: BusinessRoleTrace

— Corresponding UML Element: corresponds to the binding of an instance of a <<BusinessRole>>-stereotyped Actor to a Lifeline in a <<BusinessTransaction>>-stereotyped Interaction.

— Mappings of the Metaclass's Properties:

  businessRole: points to a <<BusinessRole>>-stereotyped Actor;

  participant: points to a <<BusinessTransaction>>-stereotyped Interaction.

5.5 UML realization of ISO20022::Metamodel::ConceptualLevel::Dynamic

5.5.1 General

This subclause defines how the elements of the ISO20022::Metamodel::ConceptualLevel::Dynamic Package map to UML.

6) As specified below, the Metamodel's Participant Metaclass maps to UML's Lifeline element.

7) In the Metamodel, this relationship is captured via the BusinessRoleTrace Metaclass, which relates BusinessRoles to Participants.

8) UML does not define a visual representation of the link between a UseCase and an Interaction, but UML tools typically provide a way to create the link via property sheets or model explorer windows.

9) The UML Metamodel's "ownedBehavior" Property is an end of an Association that is not navigable in the other direction; in other words, that Association does not have a backpointer that points from an Interaction back to the owning UseCase. However, the "context" Property that Interaction inherits from Behavior can be used to navigate back to the owning UseCase.

5.5.2 Metaclass: BusinessTransaction

— Corresponding UML Element: a <<BusinessTransaction>>-stereotyped Interaction.

— Mappings of the Metaclass's Properties:

businessProcessTrace: corresponds to the "context" Property that UML Interaction inherits from Behavior, pointing to the <<BusinessProcess>>-stereotyped UseCase that owns the <<BusinessTransaction>>-stereotyped Interaction;

messageTransportMode: captured via the Tag Definitions defined for the <<BusinessTransaction>> Stereotype;

parentTransaction: if "parentTransaction" points to the outermost BusinessTransaction, then the "parentTransaction" Property corresponds to the InteractionFragment's "enclosingInteraction" Property; otherwise, it corresponds to the InteractionFragment's "enclosingOperand" Property;

participant: corresponds to UML Interaction's "lifeline" Property, pointing to the <<BusinessTransaction>>-stereotyped Interaction's <<Participant>>-stereotyped Lifelines;

subTransaction: corresponds to Interaction's "fragment" Property;

trace: not implemented by the Profile. This Property is an artefact of the mapping of the Zachman Logical level to the Physical level, which is out of scope for this Profile;

transmission: corresponds to UML Interaction's "message" Property, which points to a <<MessageTransmission>>-stereotyped Message.

5.5.3 Metaclass: MessageTransmission

— Corresponding UML Element: a <<MessageTransmission>>-stereotyped Message.

— Mappings of the Metaclass's Properties:

— businessTransaction: corresponds to UML Message's "interaction" Property, which points to the <<BusinessTransaction>>-stereotyped Interaction that owns the <<MessageTransmission>>-stereotyped Message;

— derivation: points indirectly to a <<MessageDefinition>>-stereotyped Signal. The indirect path by which it points to the Signal is as follows: the Message points to a MessageOccurrenceSpecification (a subclass of MessageEnd) via Message's "receiveEvent" Property. The MessageOccurrenceSpecification, in turn, points to a SendSignalEvent via MessageOccurrenceSpecification's "event" Property. The SendSignalEvent, in turn, points to a Signal via SendSignalEvent's "signal" Property[10];

— messageTypeDescription: captured via <<MessageTransmission>>'s "messageTypeDescription" Tag Definition;

— receive: corresponds to UML Message's "receiveEvent" Property[11];

— send: corresponds to UML Message's sendEvent Property.

10) Since in UML the binding of a Message to a Signal requires the instantiation of a number of intertwined associations, UML tools usually provide a shortcut through the user interface that enables the modeller to effect the binding without having to consciously instantiate each association.

11) Although the Metamodel's "receive" Property has multiplicity 0..*, UML Message's "receiveEvent" Property has multiplicity 0..1, which introduces a restriction when using ISO 20022 with UML that is not conceptually an ISO 20022 restriction.

5.5.4 Metaclass: MessageTransportMode

— Corresponding UML Element: the Profile realizes the Properties of MessageTransportMode as Tag Definitions of the <<BusinessTransaction>> Stereotype. In effect, the Profile merges the BusinessTransaction and MessageTransportMode Metaclasses into one UML element, namely a <<BusinessTransaction>>-stereotyped Interaction.

— Mappings of the Metaclass's Properties:

  — boundedCommunicationDelay: captured via <<BusinessTransaction>>'s "boundedCommunicationDelay" TagDefinition;

  — businessTransaction: not implemented in the Profile[12)];

  — deliveryAssurance: captured via <<BusinessTransaction>>'s "deliveryAssurance" Tag Definition;

  — durability: captured via <<BusinessTransaction>>'s "durability" Tag Definition;

  — maximumClockVariation: captured via <<BusinessTransaction>>'s "maximumClockVariation" Tag Definition;

  — maximumMessageSize: captured via <<BusinessTransaction>>'s "maximumMessageSize" Tag Definition;

  — messageCasting: captured via <<BusinessTransaction>>'s "messageCasting" Tag Definition;

  — messageDeliveryOrder: captured via <<BusinessTransaction>>'s "messageDeliveryOrder" Tag Definition;

  — messageDeliveryWindow: captured via <<BusinessTransaction>>'s "messageDeliveryWindow" Tag Definition;

  — messageSendingWindow: captured via <<BusinessTransaction>>'s "messageSendingWindow" Tag Definition;

  — messageValidationLevel: captured via <<BusinessTransaction>>'s "messageValidationLevel" Tag Definition;

  — messageValidationOnOff: captured via <<BusinessTransaction>>'s "messageValidationOnOff" Tag Definition;

  — messageValidationResults: captured via <<BusinessTransaction>>'s "messageValidationResults" Tag Definition;

    — receiverAsynchronicity: captured via a <<BusinessTransaction>>'s "receiverAsynchronicity" Tag Definition;

    — senderAsynchronicity: captured via a <<BusinessTransaction>>'s "senderAsynchronicity" Tag Definition.

5.5.5 Metaclass: Participant

— Corresponding UML Element: a <<Participant>>-stereotyped Lifeline that is part of a <<BusinessTransaction>>-stereotyped Interaction.

12) The "businessTransaction" Property is not implemented because, as noted above, the Profile merges the BusinessTransaction and MessageTransportMode Metaclasses into one UML element, namely a <<BusinessTransaction>>-stereotyped Interaction.

— Mappings of the Metaclass's Properties:

  — businessRoleTrace: corresponds to the binding of an instance of a <<BusinessRole>>-stereotyped Actor to a Lifeline in an Interaction; such use sets this Property to point to the Actor;

  — businessTransaction: corresponds to Lifeline's "interaction" Property, pointing to the <<BusinessTransaction>>-stereotyped Interaction that encloses the <<Participant>>-stereotyped Lifeline;

  — cardinality: captured via <<Participant>>'s "minimumOccurrence" and "maximumOccurrence" Tag Definitions;

  — receives: corresponds to the "receiveEvent" Property of a <<MessageTransmission>>-stereotyped Message for which the <<Participant>>-stereotyped Lifeline is the "receiver";

  — sends: corresponds to the "sendEvent" Property of a <<MessageTransmission>>-stereotyped Message for which the <<Participant>>-stereotyped Lifeline is the "sender".

5.5.6 Metaclass: Receive

— Corresponding UML Element: the "receiveEvent" Property of a <<MessageTransmission>>-stereotyped UML Message.

— Mappings of the Metaclass's Properties:

  — messageTransmission: corresponds to the <<MessageTransmission>>-stereotyped Message that causes the receiveEvent;

  — receiver: corresponds to the <<Participant>>-stereotyped Lifeline that, in effect, receives the <<MessageTransmission>>-stereotyped Message.

5.5.7 Metaclass: Send

— Corresponding UML Element: the "sendEvent" Property of a <<MessageTransmission>>-stereotyped UML Message.

— Mappings of the Metaclass's Properties:

  — messageTransmission: corresponds to the <<MessageTransmission>>-stereotyped Message that causes the sendEvent;

  — sender: the <<Participant>>-stereotyped Lifeline that, in effect, sends the <<MessageTransmission>>-stereotyped Message.

5.6 UML realization of ISO20022::Metamodel::ConceptualLevel::Static

5.6.1 General

This Package defines how the elements of the ISO20022::Metamodel::ConceptualLevel::Static Package map to UML.

5.6.2 Metaclass: BusinessAssociation

— Corresponding UML Element: an Association whose ends are <<BusinessElement>>-stereotyped Properties.

— Mappings of the Metaclass's Properties:

— endA: corresponds to the first element of an instance of UML Association's "navigableOwnedEnd" multivalued Property, which is an ordered collection;

— endB: corresponds to the second element of an instance of UML Association's "navigableOwnedEnd" multivalued Property.

5.6.3 Metaclass: BusinessAssociationEnd

— Corresponding UML Element: a <<BusinessElement>>-stereotyped Property that is the memberEnd of an Association.

— Mappings of the Metaclass's Properties:

— type: corresponds to UML Property's "type" Property;

— aggregation: corresponds to UML Property's "aggregation" Property (see 5.11.1, which defines how the Metamodel's Aggregation Enumeration maps to UML's AggregationKind Enumeration);

— associationForEndB: corresponds to UML Property's "owningAssociation" Property[13];

— associationForEndA: corresponds to UML Property's "owningAssociation" Property.

5.6.4 Metaclass: BusinessAttribute

— Corresponding UML Element: a <<BusinessElement>>-stereotyped Property that is not the memberEnd of an Association.

— Mappings of the Metaclass's Properties:

— simpleType: corresponds to UML Property's "type" Property, pointing to a DataType;

— complexType: corresponds to UML Property's "type" Property, pointing to a <<BusinessComponent>>-stereotyped Class.

NOTE These four Properties of BusinessAttribute all have 0..1 multiplicity in the Metamodel, and there is a Constraint that says that exactly one of them is not empty. Thus, the fact that a UML Property has only a single "type" Property does not conflict with the fact that four of BusinessAttribute's Properties are mapped to the single UML "type" Property.

5.6.5 Metaclass: BusinessComponent

— Corresponding UML Element: a <<BusinessComponent>>-stereotyped Class[14].

— Mappings of the Metaclass's Properties:

— derivation: corresponds to the "supplierDependency" memberEnd of the Association between UML's NamedElement and Dependency Metaclasses (the other memberEnd of the Association is named "supplier"). The <<BusinessComponent>>-stereotyped Class is the NamedElement playing the "supplier" role and a <<Trace>>-stereotyped Dependency plays the "supplierDependency" role (a <<MessageComponentType>>-stereotyped Class is the client end of the Dependency);

13) The reason that "associationForEndA" and "associationForEndB" can both correspond to UML Property's "owningAssociation" Property, even though "owningAssociation" is a single-valued Property, is that the Metamodel specifies that exactly one of "associationForEndA" and "associationForEndB" is non-empty. Note also that the BusinessAssociationEnd is playing the "endB" role, and points back to its owning BusinessAssociation via the "associationForEndB" Property.

14) UML Instance diagrams can be used to model M0 example instances of <<BusinessComponent>>-stereotyped Classes.

— element: corresponds to the “ownedAttribute” Property of UML Class, pointing to the <<BusinessComponent>>-stereotyped Class’s Properties;

— associationDomain: corresponds to the non-navigable memberEnd of the Association between UML TypedElement and Type, where the non-navigable end is named “type”. The <<BusinessComponent>>-stereotyped Class is the Type and the TypedElement is the union of all “navigableOwnedEnds” for which the <<BusinessComponent>-stereotyped Class is the Type;

— subType: corresponds to the "specific" memberEnd of the composition association between UML's Classifier and Generalization Metaclasses, where the <<BusinessComponent>>-stereotyped class is the Classifier;

— superType: corresponds to the "general" memberEnd of the Association between UML's Classifier and Generalization Metaclasses, where the <<BusinessComponent>>-stereotyped Class is one of the Classifiers belonging to the "specific" memberEnd of the Generalization considered.

5.6.6 Metaclass: BusinessElement

— Corresponding UML Element: a << BusinessElement>>-stereotyped Property.

— Mappings of the Metaclass’s Properties:

— derivation: corresponds to the “supplierDependency” memberEnd of the Association between UML’s NamedElement and Dependency Metaclasses (the other memberEnd of the Association is named “supplier”). The <<BusinessElement>>-stereotyped Property is the NamedElement playing the “supplier” role and a <<Trace>>-stereotyped Dependency plays the “supplierDependency” role (a <<MessageElement>>-stereotyped Property is the client end of the Dependency);

— elementContext: corresponds to UML Property’s “class” Property, pointing to the <<BusinessComponent>>-stereotyped Class that owns the <<BusinessElement>>-stereotyped Property;

— cardinality: corresponds to Properties of the UML Metaclass MultiplicityElement that UML’s Property Metaclass inherits (see above, regarding the Metamodel’s Cardinality DataType for details about the correspondence between Properties of Cardinality and Properties of MultiplicityElement);

— isDerived: corresponds to UML Property’s "isDerived" Property.

5.7 UML realization of ISO20022::Metamodel::ConceptualToLogicalTransformation

5.7.1 General

This subclause defines how the elements of the ISO20022::Metamodel::ConceptualToLogicalTransformation Package map to UML.

NOTE In a UML Dependency in which element B depends on element A, element A is called the “supplier” and element B is called the “client”.

5.7.2 Metaclass: BusinessComponentTrace

— Corresponding UML Element: a <<Trace>>-stereotyped Dependency of a <<MessageComponentType>>-stereotyped Class on a <<BusinessComponent>>-stereotyped Class, or a <<Trace>>-stereotyped Dependency of a <<MessageElement>>-stereotyped Property on a <<BusinessComponent>>-stereotyped Class.

— Mappings of the Metaclass’s Properties:

- businessComponent: corresponds to the supplier end of a <<Trace>>-stereotyped Dependency of a <<MessageComponent>>-stereotyped Class on a <<BusinessComponent>>-stereotyped Class, or of a <<MessageElement>>-stereotyped Property after <<MessageComponent>>-stereotyped Class;

- messageComponentType: corresponds to the client end of a <<Trace>>-stereotyped Dependency of a <<MessageComponentType>>-stereotyped Class on a <<BusinessComponent>>-stereotyped Class;

- messageElement: corresponds to the client end of a <<Trace>>-stereotyped Dependency of a <<MessageElement>>-stereotyped Property on a <<BusinessComponent>>-stereotyped Class.

5.7.3 Metaclass: BusinessElementTrace

- Corresponding UML Element: a <<Trace>>-stereotyped Dependency of a <<MessageElement>>-stereotyped Property on a <<BusinessElement>>-stereotyped Property.

- Mappings of the Metaclass's Properties:

  - businessElement: corresponds to the supplier end of the Dependency;

  - messageElement: corresponds to the client end of the Dependency.

5.7.4 Metaclass: CodeSetTrace

- Corresponding UML Element: a <<Trace>>-stereotyped Dependency of a <<CodeSet>>-stereotyped Enumeration on another <<CodeSet>>-stereotyped Enumeration.

- Mappings of the Metaclass's Properties:

  derivedCodeSet: corresponds to the client end of the Dependency;

  sourceCodeSet: corresponds to the supplier end of the Dependency.

5.7.5 Metaclass: MessageTypeTrace

- Corresponding UML Element: corresponds to the binding of a <<MessageTransmission>>-stereotyped Message to a <<MessageDefinition>>-stereotyped Signal.

- Mappings of the Metaclass's Properties:

  messageDefinition: the Profile does not implement this Property[15];

  messageTransmission: points to a <<MessageTransmission>-stereotyped Message[16] (see 5.5.3, which defines how a <<MessageTransmission>>-Stereotyped Message binds to a <<MessageDefinition>>-Stereotyped Signal via instances of Associations of UML's Metamodel).

15) One would expect to realize this Property by using instances of the same Associations of the UML Metamodel that the Profile uses to realize the "messageTransmission" Property. That would be the expectation because "messageDefinition" points a MessageTransmission to a MessageDefinition, whereas "messageTransmission" points a MessageDefinition to a MessageTransmission; in essence, "messageDefinition" is the inverse of "messageTransmission". However, as explained in 5.5.3, the Associations that UML uses to point a Message to a Signal are unidirectional, and thus do not provide a path to navigate from the Signal to the Message.

16) In 5.4.2, the Associations that UML uses to point a Message to a Signal are unidirectional, and thus do not provide a path to navigate from the Signal to the Message.

5.8 UML realization of ISO20022::Metamodel::LogicalLevel

5.8.1 General

This subclause defines how the elements of the ISO20022::Metamodel::LogicalLevel Package map to UML.

5.8.2 Metaclass::BusinessArea

— Corresponding UML Element: a <<BusinessArea>>-stereotyped Class.

— Mappings of the Metaclass's Properties:

  — messageDefinition: corresponds to the component memberEnd of a composition Association between the <<BusinessArea>>-stereotyped Class (the composite) and a <<MessageDefinition>>-stereotyped Signal (the component);

  — code: captured via a <<BusinessArea>> "code" Tag Definition.

5.8.3 Metaclass: ChoiceComponent

— Corresponding UML Element: a <<ChoiceComponent>>-stereotyped Class.

— Mappings of the Metaclass's Properties: this Metaclass has no Properties.

5.8.4 Metaclass: ExternalSchema

— Corresponding UML Element: an <<ExternalSchema>>-stereotyped Class.

— Mappings of the Metaclass's Properties:

  — processContent: captured via an <<ExternalSchema>>'s "processContent" Tag Definition;

  — namespaceList: captured via an <<ExternalSchema>>'s "namespaceList" Tag Definition.

5.8.5 Metaclass: MessageAssociation

— Corresponding UML Element: an Association, the ends of which are <<MessageElement>>-stereotyped Properties.

— Mappings of the Metaclass's Properties:

  — source: corresponds to the first element of an instance of UML Association's "navigableOwnedEnd" multivalued Property;

  — target: corresponds to the second Element of an instance of UML Association's "navigableOwnedEnd" multivalued Property.

5.8.6 Metaclass: MessageAssociationEnd

— Corresponding UML Element: a <<MessageElement>>-stereotyped Property that is the memberEnd of an Association.

— Mappings of the Metaclass's Properties:

- associationForSource: corresponds to the Property's "owningAssociation" Property[17];
- associationForTarget: corresponds to the Property's "owningAssociation" Property;
- isComposite: corresponds to the Property's "aggregation" Property. An isComposite value of "true" corresponds to the UML's AggregationKind::composite EnumerationLiteral. An isComposite value of "false" corresponds to AggregationKind::none;
- type: corresponds to the Property's "type" Property.

5.8.7 Metaclass: MessageAttribute

- Corresponding UML Element: a <<MessageElement>>-stereotyped Property that is not the memberEnd of an Association.
- Mappings of the Metaclass's Properties:
  - complexType: corresponds to the Property's "type" Property, pointing to a <<MessageComponentType>>-stereotyped Class;
  - simpleType: corresponds to the Property's "type" Property, pointing to a DataType.

NOTE These four Properties of MessageAttribute all have 0..1 multiplicity in the Metamodel, and there is a Constraint that says that exactly one of them is not empty. Thus, the fact that a UML Property has only a single "type" Property does not conflict with the fact that four of MessageAttribute's Properties are mapped to the single UML "type" Property.

5.8.8 Metaclass: MessageBuildingBlock

- Corresponding UML Element: a <<MessageBuildingBlock>>-stereotyped Property that is a memberEnd of an Association between a <<MessageDefinition>>-stereotyped Signal and a <<MessageComponentType>>-stereotyped Class. The memberEnd is the one on the <<MessageComponentType>>-stereotyped Class's side of the Association. The Association in question is referred to as "the Association" for the remainder of 5.8.8.
- Mappings of the Metaclass's Properties:
  - type: corresponds to the <<MessageComponentType>>-stereotyped Class that is a memberEnd of the Association;
  - cardinality: corresponds to the cardinality of the <MessageBuildingBlock>>-stereotyped Property.

NOTE In the Metamodel, the Association between MessageBuildingBlock and MessageDefinition is not navigable on the MessageDefinition memberEnd. That is why the Metamodel contains no Property corresponding to that memberEnd.

5.8.9 Metaclass: MessageComponent

- Corresponding UML Element: a << MessageComponent>>-stereotyped Class.
- Mappings of the Metaclass's Properties: this Metaclass has no Properties.

17) The reason that "associationForTarget" and "associationForSource" can both correspond to UML Property's "owningAssociation" Property, even though "owningAssociation" is a single-valued Property, is that the Metamodel specifies that exactly one of "associationForSource" and "associationForTarget" is non-empty.

5.8.10 Metaclass: MessageComponentType

— Corresponding UML Element: a Class that is stereotyped by one of the concrete subtypes of <<MessageComponentType>>[18].

— Mappings of the Metaclass's Properties:

  — isTechnical: captured via a <<MessageComponentType>>'s "isTechnical" Tag Definition;

  — messageElement: corresponds to the "ownedAttribute" Property of UML Class;

  — trace: corresponds to the "clientDependency" memberEnd of the Association between UML's NamedElement and Dependency Metaclasses (the other memberEnd of the Association is named "client"). The <<MessageComponentType>>-stereotyped Class is the NamedElement playing the "client" role and a <<Trace>>-stereotyped Dependency plays the "clientDependency" role (a <<BusinessComponent>>-stereotyped Class is the supplier end of the Dependency);

  — messageBuildingBlock: corresponds to an Association between the <<MessageComponentType>>-stereotyped Class and a <<MessageDefinition>>-stereotyped Signal.

5.8.11 Metaclass: MessageDefinition

— Corresponding UML Element: a << MessageDefinition>>–stereotyped signal[19].

— Mappings of the Metaclass's Properties:

  — businessArea: corresponds to the composite memberEnd of a composition Association between the <<MessageDefinition>>-stereotyped Signal (the component) and a <<BusinessArea>>-stereotyped Class (the composite);

  — derivation: not implemented by the Profile, because this Property relates to the Physical level[20];

  — messageBuildingBlock: corresponds to a memberEnd of an Association between the <<MessageDefinition>>-stereotyped Signal and a <<MessageComponentType>>-stereotyped Class. The memberEnd is the one on the <<MessageComponentType>>-stereotyped Class's side of the Association and the memberEnd is stereotyped <<MessageBuildingBlock>>;

  — messageSchema: not covered by the Profile because this is a Physical level element;

  — messageSet: corresponds to the <<Message-Set>>-stereotyped Artefacts that own a UML Manifestation in which this <<MessageDefinition>>-stereotyped Class is the utilizedElement property;

  — trace: not implemented in the Profile. In principle, this Property points to a <<MessageTransmission>>-stereotyped Message, via the inverse of the path by which a Message points to a Signal; that path is defined in 5.5.3, in the definition of the Profile's implementation of MessageTransmission's "derivation" Property. However, the Associations that are instantiated to construct that path are unidirectional, so the links that instantiate the Associations do not point the Signal back to the Message;

18) UML Instance diagrams can be used to model M0 example instances of <<MessageComponentType>>-stereotyped Classes.

19) UML Instance diagrams can be used to model M0 example instances of <<MessageDefinition>>-stereotyped Signals

20) As mentioned earlier, the Profile covers only the Scope, Conceptual, and Logical levels. Metamodel elements relating to the Physical level are not implemented in the Profile. In the Metamodel, this "derivation" Property points the MessageDefinition to a SyntaxMessageScheme. The SyntaxMessageScheme is part of the Physical level.

- messageDefinitionIdentifier: captured via “messageDefinitionIdentifier” Tag Definition of <<MessageDefinition>>. Although the Metamodel dictates that an instance of the “messageDefinitionIdentifier” Property references an instance of the MessageDefinitionIdentifier Metaclass, in the Profile an instance of MessageDefinitionIdentifier collapses into one string that encodes the values of MessageDefinitionIdentifier’s four Properties. This string is captured in the “messageDefinitionIdentifier” Tag Definition. The format of this string is specified in ISO 20022-3;

- rootElement: captured via <<MessageDefinition>>'s “rootElement” Tag Definition[21)].

5.8.12 Metaclass: MessageDefinitionIdentifier

- Corresponding UML Element: the four Properties of MessageDefinitionIdentifier (businessArea, messageFunctionality, flavour, and version) are encoded in <<MessageDefinition>> “messageDefinitionIdentifier” Tag Definition (see 5.8.11 for mapping of the MessageDefinition Metaclass’s “messageDefinitionIdentifier” Property)[22)].

- Mappings of the Metaclass’s Properties: see the explanation in the previous paragraph.

5.8.13 Metaclass: MessageElement

- Corresponding UML Element: a << MessageElement>>–stereotyped Property.

- Mappings of the Metaclass’s Properties:

  - isTechnical: captured via <<MessageElement>>'s “isTechnical” Tag Definition;

  - businessElementTrace: corresponds to the “clientDependency” memberEnd of the Association between UML’s NamedElement and Dependency Metaclasses (the other memberEnd of the Association is named “client”). The <<MessageElement>>-stereotyped Property is the NamedElement playing the “client” role and a <<Trace>>-stereotyped Dependency plays the “clientDependency” role (a <<BusinessElement>>-stereotyped Property is the supplier end of the Dependency);

  - cardinality: corresponds to Properties that UML’s Property Metaclass inherits from MultiplicityElement (see 5.2.13 regarding the Metamodel’s Cardinality DataType for details about relevant inherited Properties);

  - componentContext: corresponds to UML Property’s “class” Property, which points to the <<MessageComponentType>>-stereotyped Class that owns the <<MessageElement>>-stereotyped Property;

  - isDerived: corresponds to UML Property’s "isDerived" Property;

  - businessComponentTrace: corresponds to the “clientDependency” memberEnd of the Association between UML’s NamedElement and Dependency Metaclasses (the other memberEnd of the Association is named “client”). The <<MessageElement>>-stereotyped Property is the NamedElement playing the “client” role and a <<Trace>>-stereotyped Dependency plays the “clientDependency” role (a <<BusinessComponent>>-stereotyped Property is the supplier end of the Dependency).

21) ISO 20022 traditionally names the root element “Document”, but this Tag Definition makes it possible to designate a different root element name such as “FpML”.

22) In essence, the Profile flattens the model, merging MessageDefinitionIdentifier into MessageDefinition. The Metamodel specifies that an instance of the MessageDefinition Metaclass contains an instance of the MessageDefinitionIdentifier Metaclass, via composition. In the Profile, the instance of MessageDefinitionIdentifier is implemented as an instance of <<MessageDefinition>>’s “messageDefinitionIdentifier” Tagged Value.

5.8.14 Metaclass: MessageSet

— Corresponding UML Element: a << MessageSet>>–stereotyped Artefact.

— Mappings of the Metaclass's Properties:

  — messageDefinition: corresponds to the utilizedElement UML Property of the UML Manifestation instance linked to the <<MessageSet>>-stereotyped Artefact via the manifestation UML Property.

5.9 UML realization of ISO20022::Metamodel::DataTypes

5.9.1 General

For each DataType that is an element of the ISO20022::Metamodel::DataTypes Package, the Profile defines a corresponding Stereotype with the same unqualified name. Each of these Stereotypes extends the UML DataType Metaclass, each is contained in the ISO20022::Profile::DataTypes Package, and each is defined in Annex A.

5.9.2 DataType: Amount

— Corresponding UML Element: an <<Amount>>-stereotyped DataType.

— Implementation of the DataType's Properties:

  — currencyIdentification: captured via <<Amount>>'s "currencyIdentification" Tag Definition.

5.9.3 DataType: Indicator

— Corresponding UML Element: an <<Indicator>>-stereotyped DataType.

— Implementation of the DataType's Properties:

  — meaningWhenTrue: captured via <<Indicator>>'s "meaningWhenTrue" Tag Definition;

  — meaningWhenFalse: captured via <<Indicator>>'s "meaningWhenFalse" Tag Definition.

5.9.4 DataType: Quantity

— Corresponding UML Element: a <<Quantity>>-stereotyped DataType.

— Implementation of the DataType's Properties:

  — unitCode: captured via a <<Quantity>>'s "unitCode" Tag Definition.

5.9.5 DataType: Rate

— Corresponding UML Element: a <<Rate>>-stereotyped DataType.

— Implementation of the DataType's Properties:

  — baseValue: captured via a <<Rate>>'s "baseValue" Tag Definition;

  — baseUnitCode: captured via a <<Rate>>'s "baseUnitCode" Tag Definition.

5.9.6 DataType: Text

— Corresponding UML Element: a <<Text>>-stereotyped DataType;

— Implementation of the DataType's Properties: this DataType has no Properties.

5.10 UML realization of ISO20022::TypeLibrary::XMLSchema

— The Profile imports the ISO20022::TypeLibrary::XMLSchema Package[23]. ISO 20022-compliant UML models may use only a subset of the DataTypes that the Package contains. The subset consists of the following DataTypes:

  — base64Binary;

  — boolean;

  — date;

  — dateTime;

  — decimal;

  — duration;

  — gDay;

  — gMonth;

  — gMonthDay;

  — gYear;

  — gYearMonth;

  — integer;

  — string;

  — time.

— This subset of fourteen elements constitutes the set of basic DataTypes that are built into the Profile. ISO 20022-compliant models may use any element of this subset as the type of a <<BusinessElement>>-stereotyped UML Attribute or as the type of a <<MessageElement>>-stereotyped UML Attribute[24].

— The Profile also defines (in Annex A) a corresponding Stereotype of the UML DataType Metaclass. These Stereotypes are contained in the ISO20022::Profile::DataTypes Package. The purpose of these Stereotypes is to provide UML modellers with a means to define new DataTypes that are derived from the Profile's built-in DataTypes. Thus, we refer to these Stereotypes as the Restriction Stereotypes. Each of the Restriction Stereotypes defines and/or inherits Tag Definitions that correspond to XML Schema Constraining Facets. The semantics of each of these Tag Definitions are the same as the semantics of the corresponding XML Schema Constraining Facet. The following is a list of the Restriction Stereotypes (the Restriction Stereotype that corresponds to a given built-in Profile DataType is in the same position in this list as the position of the corresponding built-in DataType in the list of the fourteen built-in DataTypes above):

23) The ISO20022::TypeLibrary::XMLSchema Package, defined in ISO 20022-1, contains definitions of the W3C XML Schema built-in Datatypes in a form that makes it possible for modellers to use these Datatypes in UML and MOF models.

24) The Metamodel and Profile do not restrict themselves to this subset for defining the Metamodel and for the defining Profile. Thus, there are Properties of the Metamodel and Tag Definitions in the Profile whose types are members of the ISO20022::TypeLibrary::XMLSchema and are not members of the subset.

- <<Binary>>
- <<Indicator>>
- <<Date>>
- <<DateTime>>
- <<Decimal>>
- <<Duration>>
- <<Day>>
- <<Month>>
- <<MonthDay>>
- <<Year>>
- <<YearMonth>>
- <<Text>>
- <<Time>

- Thus, for example, a modeller who wishes to define by restriction a new DataType based on the built-in month DataType shall stereotype the new DataType <<Month>> and use at least one of the defined or inherited Tag Definitions of <<Month>> to constrain the values of instances of the new DataType. For each of the Tag Definitions of the Restriction Stereotypes, the XML Schema Constraining Facet to which the Tag Definition corresponds can be inferred from the fact that the unqualified name of the Tag Definition is the same as the name of the corresponding XML Schema Constraining Facet.

NOTE The fact that the Profile uses a subset of the XML Schema built-in types as its set of built-in DataTypes does not mean that the Profile and models created with the Profile are XML-specific. There is nothing inherently XML-specific about the XML built-in DataTypes; the W3C XML Schema specification defines many of them in terms of pre-existing IEEE and ISO standards. Some set of primitive types had to be chosen for the Profile, and the XML Schema DataTypes have the advantage of being well known and widely supported in the computer industry. A UML model that conforms to the Profile can be used to generate not only XML as specified by ISO 20022-4, but potentially also can be used to generate other kinds of artefacts, such as EDIFACT-based or JSON-based definitions.

5.11 UML realization of ISO20022::TypeLibrary::Enumerations

As explained in 4.2, the Profile imports and reuses the ISO20022::TypeLibrary::Enumerations Package. There is no need for this subclause of the Profile Specification to describe each Enumeration. The Profile simply uses these Enumerations (as the types of Tag Definitions).

However, the Profile does not actually use the Aggregation Enumeration, which is contained in the Enumerations Package, because UML has a nearly identical Enumeration. The remainder of this subclause is devoted to explaining how the Profile maps the Aggregation Enumeration to the corresponding UML Enumeration.

5.11.1 Enumeration: Aggregation

- Corresponding UML Element: AggregationKind.

- Mappings of the Enumeration's EnumerationLiterals: Aggregation's EnumerationLiterals correspond exactly by name and by semantics to the EnumerationLiterals of UML's AggregationKind Enumeration. Thus an Aggregation value is set by setting the UML AggregationKind value as desired.

Annex A
(normative)

Definitions of Stereotypes and Tags

A.1 General

This Annex defines the Profile's Stereotypes and Tag Definitions, to which previous sections refer. Each major subsection corresponds to one of the Profile's Packages, and contains the definitions of the Stereotypes contained by that Package and the Tag Definitions for those Stereotypes. The standard UML <<profile>> Stereotype is applied to each Package. At the beginning of the subsection for each Package, there is a non-normative UML Class diagram of the Stereotypes and Tag Definitions. All of the Stereotypes defined by the Profile have public visibility.

When the Profile defines a Tag, it assigns values to the UML Properties of each Tag Definition[25]. For some of those UML Properties, the Profile defines the same value for all of the Tag Definitions. Instead of listing those Properties and their values in line within each Tag Definition, Table B.1 lists these Properties and their values once:

Table B.1 — Fixed values of Tag Definitions' UML Properties

Property	Value for all Tag Definitions
aggregation	none
isDerived	false
isReadOnly	false
isStatic	false
isUnique	true
visibility	public

Figure A.1 illustrates the internal structure of the ISO20022::Profile Package and shows the Packages that it imports.

A.2 Stereotype and Tag details

A.2.1 Package ISO20022

Applied Stereotype	

25) In UML, Tag Definitions are instances of the UML Property Metaclass. They are Properties of their owning Stereotype. Therefore, the Properties listed in Table 1 are Properties of UML's Property Metaclass.

Description

A.2.2 Package ISO20022::Profile

Applied Stereotype	<<profile>>

Description

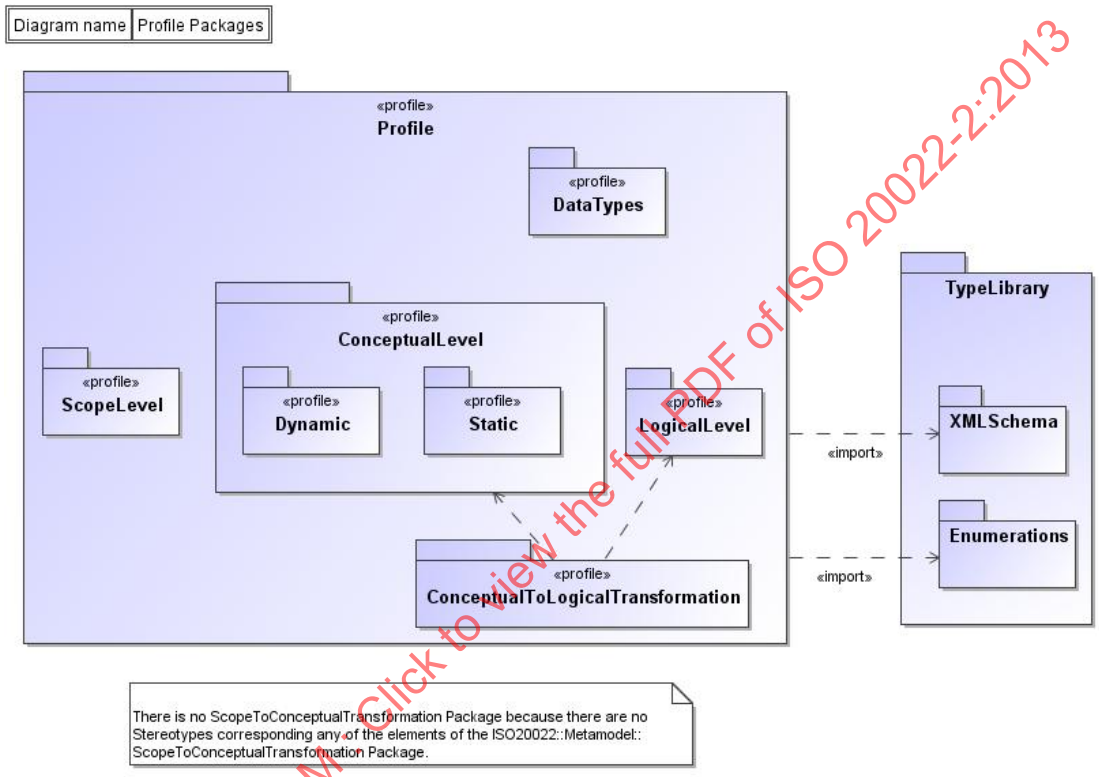

Figure A.1 — Example of Profile Packages

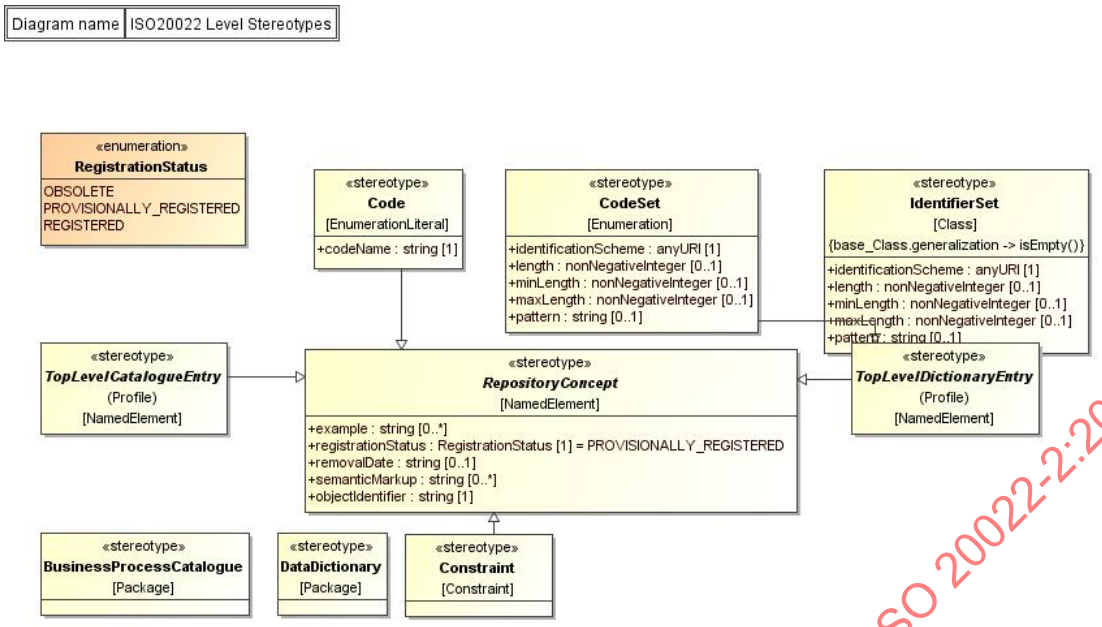

Figure A.2 — Example of ISO20022 Level Stereotypes

A.2.2.1 Stereotype BusinessProcessCatalogue

Description

Qualified name	ISO20022::Profile::BusinessProcessCatalogue
Abstract	false
Owner	ISO20022::Profile
Base UML Metaclass	Package
Superclass	

Constraints

— BusinessProcessCatalogueLegalOwner

Description

A <<BusinessProcessCatalogue>>-stereotyped Package can be owned only by a <<BusinessProcessCatalogue>>-stereotyped Package.

Context	ISO20022::Profile::BusinessProcessCatalogue
Language	OCL2.0
Body	not(base_Package.nestingPackage.oclIsUndefined()) implies base_Package.nestingPackage.oclIsTypeOf(BusinessProcessCatalogue)

— BusinessProcessCatalogueLegalToOwn.

Description

All sub-Packages of a <<BusinessProcessCatalogue>>--stereotyped Package shall be <<BusinessProcessCatalogue>>-stereotyped.

Context	ISO20022::Profile::BusinessProcessCatalogue
Language	OCL2.0
Body	base_Package.nestedPackage -> forAll (pkg | pkg.oclIsTypeOf(BusinessProcessCatalogue))

A.2.2.2 Stereotype Code

Description

Qualified name	ISO20022::Profile::Code
Abstract	false
Owner	ISO20022::Profile
Base UML Metaclass	EnumerationLiteral
Superclass	ISO20022::Profile::RepositoryConcept

Tag Definitions

— codeName

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::Code
Default Value	
Multiplicity	1
Ordered	false

Constraints

— MustBeLiteralOfCodeSetEnumeration

Description

A <<Code>>-stereotyped EnumerationLiteral shall be part of a <<CodeSet>>-stereotyped Enumeration.

Context	ISO20022::Profile::Code
Language	OCL2.0
Body	base_EnumerationLiteral.owner.oclIsTypeOf(CodeSet)

A.2.2.3 Stereotype CodeSet

Description

Qualified name	ISO20022::Profile::CodeSet

Abstract	false
Owner	ISO20022::Profile
Base UML Metaclass	Enumeration
Superclass	ISO20022::Profile::TopLevelDictionaryEntry

Tag Definitions

— identificationScheme

Type	ISO20022::TypeLibrary::XMLSchema::anyURI
Owner	ISO20022::Profile::CodeSet
Default Value	
Multiplicity	1
Ordered	false

— length

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::CodeSet
Default Value	
Multiplicity	0..1
Ordered	false

— maxLength

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::CodeSet
Default Value	
Multiplicity	0..1
Ordered	false

— minLength

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::CodeSet
Default Value	
Multiplicity	0..1
Ordered	false

— pattern

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::CodeSet
Default Value	
Multiplicity	0..1
Ordered	false

A.2.2.4 Stereotype Constraint

Description

Qualified name	ISO20022::Profile::Constraint
Abstract	false
Owner	ISO20022::Profile
Base UML Metaclass	Constraint
Superclass	ISO20022::Profile::RepositoryConcept

Constraints

— MustBeOpaqueExpression

Description

A <<Constraint>>-stereotyped Constraint's "specification" property shall be an OpaqueExpression.

Context	ISO20022::Profile::Constraint
Language	OCL2.0
Body	base_Constraint.specification.oclIsKindOf(OpaqueExpression)

A.2.2.5 Stereotype DataDictionary

Description

Qualified name	ISO20022::Profile::DataDictionary
Abstract	false
Owner	ISO20022::Profile
Base UML Metaclass	Package
Superclass	

Constraints

— DataDictionaryLegalOwner

Description

A <<DataDictionary>>-stereotyped Package can be owned only by a <<DataDictionary>>-stereotyped Package.

Context	ISO20022::Profile::DataDictionary
Language	OCL2.0
Body	not(base_Package.nestingPackage.oclIsUndefined()) implies base_Package.nestingPackage.oclIsTypeOf(DataDictionary)

— DataDictionaryLegalToOwn

Description

All sub-Packages of a <<DataDictionary>>--stereotyped Package shall be <<DataDictionary>>-stereotyped.

Context	ISO20022::Profile::DataDictionary
Language	OCL2.0
Body	base_Package.nestedPackage -> forAll (pkg | pkg.oclIsTypeOf(DataDictionary))

A.2.2.6 Stereotype IdentifierSet

Description

Qualified name	ISO20022::Profile::IdentifierSet
Abstract	false
Owner	ISO20022::Profile
Base UML Metaclass	Class
Superclass	ISO20022::Profile::TopLevelDictionaryEntry

Tag Definitions

— identificationScheme

Type	ISO20022::TypeLibrary::XMLSchema::anyURI
Owner	ISO20022::Profile::IdentifierSet
Default Value	
Multiplicity	1
Ordered	false

— length

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::IdentifierSet
Default Value	
Multiplicity	0..1
Ordered	false

— maxLength

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::IdentifierSet
Default Value	
Multiplicity	0..1
Ordered	false

— minLength

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::IdentifierSet
Default Value	
Multiplicity	0..1
Ordered	false

— pattern

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::IdentifierSet
Default Value	
Multiplicity	0..1
Ordered	false

Constraints

— IdentifierSetSubclassingProhibited

Description

Context	ISO20022::Profile::IdentifierSet
Language	OCL2.0
Body	base_Class.generalization -> isEmpty()

A.2.2.7 Stereotype RepositoryConcept

Description

Qualified name	ISO20022::Profile::RepositoryConcept
Abstract	true
Owner	ISO20022::Profile
Base UML Metaclass	NamedElement
Superclass	

Tag Definitions

— example

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::RepositoryConcept
Default Value	
Multiplicity	0..*
Ordered	false

— objectIdentifier

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::RepositoryConcept
Default Value	
Multiplicity	1
Ordered	false

— registrationStatus

Type	ISO20022::TypeLibrary::Enumerations::RegistrationStatus
Owner	ISO20022::Profile::RepositoryConcept
Default Value	PROVISIONALLY_REGISTERED
Multiplicity	1

Ordered	false

— removalDate

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::RepositoryConcept
Default Value	
Multiplicity	0..1
Ordered	false

— semanticMarkup

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::RepositoryConcept
Default Value	
Multiplicity	0..*
Ordered	false

Constraints

— NoOperationsInClassExtensions

Description

Stereotypes extending the meta-class Class (MessageComponentType, BusinessComponent, MessageSet, IdentifierSet, BusinessArea) may not contain operations.

Context	ISO20022::Profile::RepositoryConcept
Language	OCL2.0
Body	base_NamedElement.oclIsKindOf(Class) implies base_NamedElement.oclAsType(Class).ownedOperation -> isEmpty()

— RemovalDateImplication

Description

If a removalDate is specified then the registrationStatus must be OBSOLETE.

Context	ISO20022::Profile::RepositoryConcept
Language	OCL2.0
Body	(not removalDate.oclIsUndefined()) implies registrationStatus = RegistrationStatus::OBSOLETE

A.2.2.8 Stereotype TopLevelCatalogueEntry

Description

Qualified name	ISO20022::Profile::TopLevelCatalogueEntry
Abstract	true
Owner	ISO20022::Profile
Base UML Metaclass	NamedElement
Superclass	ISO20022::Profile::RepositoryConcept

Constraints

— CatalogueEntryOwnership

Description

A <<TopLevelCatalogueEntry>>-stereotyped UML Element shall be owned by a <<BusinessProcessCatalogue>>-stereotyped Package.

Context	ISO20022::Profile::TopLevelCatalogueEntry
Language	OCL2.0
Body	owner.oclIsTypeOf(BusinessProcessCatalogue)

A.2.2.9 Stereotype TopLevelDictionaryEntry

Description

Qualified name	ISO20022::Profile::TopLevelDictionaryEntry
Abstract	true
Owner	ISO20022::Profile
Base UML Metaclass	NamedElement
Superclass	ISO20022::Profile::RepositoryConcept

Constraints

— DictionaryOwnership

Description

A <<TopLevelDictionaryEntry>>-stereotyped UML Element shall be owned by a <<DataDictionary>>-stereotyped Package.

Context	ISO20022::Profile::TopLevelDictionaryEntry
Language	OCL2.0
Body	owner.oclIsTypeOf(DataDictionary)

A.2.3 Package ISO20022::Profile::ConceptualLevel

A.2.3.1 General

Applied Stereotype	<<profile>>

A.2.4 Package ISO20022::Profile::ConceptualLevel::Dynamic

A.2.4.1 General

Applied Stereotype	<<profile>>

Description

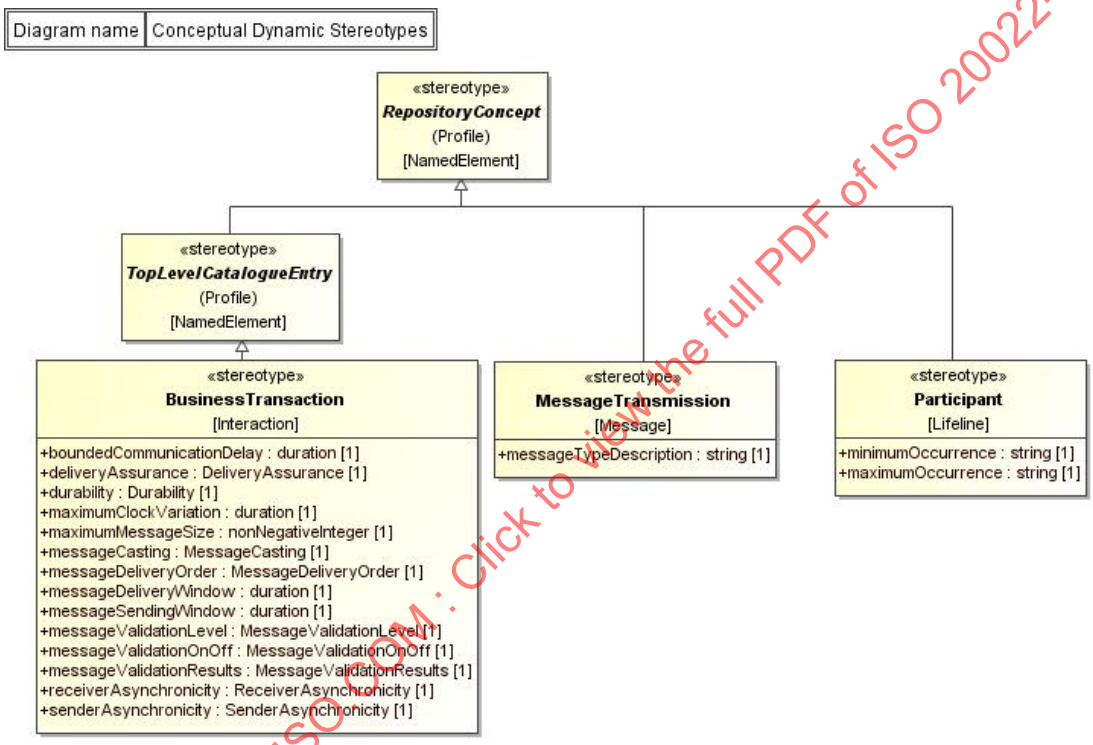

Figure A.3 — Example of conceptual Dynamic Stereotypes

A.2.4.2 Stereotype BusinessTransaction

Description

Qualified name	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Abstract	false
Owner	ISO20022::Profile::ConceptualLevel::Dynamic
Base UML Metaclass	Interaction
Superclass	ISO20022::Profile::TopLevelCatalogueEntry

Tag Definitions

— boundedCommunicationDelay

Type	ISO20022::TypeLibrary::XMLSchema::duration
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— deliveryAssurance

Type	ISO20022::TypeLibrary::Enumerations::DeliveryAssurance
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— durability

Type	ISO20022::TypeLibrary::Enumerations::Durability
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— maximumClockVariation

Type	ISO20022::TypeLibrary::XMLSchema::duration
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— maximumMessageSize

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	

Multiplicity	1
Ordered	false

— messageCasting

Type	ISO20022::TypeLibrary::Enumerations::MessageCasting
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— messageDeliveryOrder

Type	ISO20022::TypeLibrary::Enumerations::MessageDeliveryOrder
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— messageDeliveryWindow

Type	ISO20022::TypeLibrary::XMLSchema::duration
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— messageSendingWindow

Type	ISO20022::TypeLibrary::XMLSchema::duration
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— messageValidationLevel

Type	ISO20022::TypeLibrary::Enumerations::MessageValidationLevel
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— messageValidationOnOff

Type	ISO20022::TypeLibrary::Enumerations::MessageValidationOnOff
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— messageValidationResults

Type	ISO20022::TypeLibrary::Enumerations::MessageValidationResults
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— receiverAsynchronicity

Type	ISO20022::TypeLibrary::Enumerations::ReceiverAsynchronicity
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction
Default Value	
Multiplicity	1
Ordered	false

— senderAsynchronicity

Type	ISO20022::TypeLibrary::Enumerations::SenderAsynchronicity
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::BusinessTransaction

Default Value	
Multiplicity	1
Ordered	false

A.2.4.3 Stereotype MessageTransmission

Description

Qualified name	ISO20022::Profile::ConceptualLevel::Dynamic::MessageTransmission
Abstract	false
Owner	ISO20022::Profile::ConceptualLevel::Dynamic
Base UML Metaclass	Message
Superclass	ISO20022::Profile::RepositoryConcept

Tag Definitions

— messageTypeDescription

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::MessageTransmission
Default Value	
Multiplicity	1
Ordered	false

A.2.4.4 Stereotype Participant

Description

Qualified name	ISO20022::Profile::ConceptualLevel::Dynamic::Participant
Abstract	false
Owner	ISO20022::Profile::ConceptualLevel::Dynamic
Base UML Metaclass	Lifeline
Superclass	ISO20022::Profile::RepositoryConcept

Tag Definitions

— maximumOccurrence

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::Participant
Default Value	
Multiplicity	1
Ordered	false

— minimumOccurrence

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::ConceptualLevel::Dynamic::Participant
Default Value	
Multiplicity	1
Ordered	false

A.2.5 Package ISO20022::Profile::ConceptualLevel::Static

A.2.5.1 General

Applied Stereotype	<<profile>>

Description

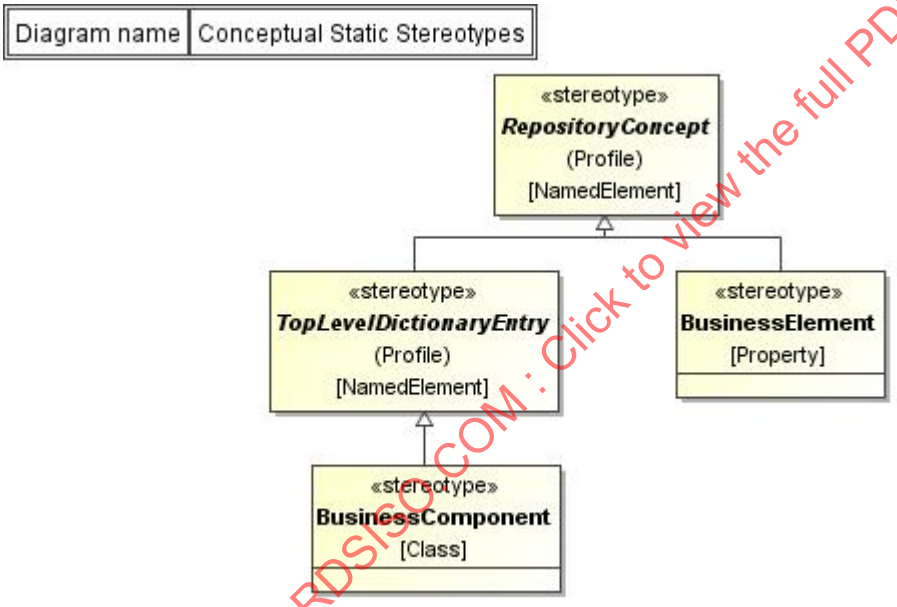

Figure A.4 — Example of Conceptual Static Stereotypes

A.2.5.2 Stereotype BusinessComponent

Description

Qualified name	ISO20022::Profile::ConceptualLevel::Static::BusinessComponent
Abstract	false
Owner	ISO20022::Profile::ConceptualLevel::Static
Base UML Metaclass	Class
Superclass	ISO20022::Profile::TopLevelDictionaryEntry

Constraints

— BusinessComponentOperationsProhibited

Description

Operations are not allowed in Business Components

Context	ISO20022::Profile::ConceptualLevel::Static::BusinessComponent
Language	OCL2.0
Body	base_Class.ownedOperation -> isEmpty()

— GeneralizationsMustBeAmongBusinessComponents

Description

If a <<BusinessComponent>>-stereotyped Class is a participant in a Generalization, the other participant must also be a <<BusinessComponent>>-stereotyped Class.

Context	ISO20022::Profile::ConceptualLevel::Static::BusinessComponent
Language	OCL2.0
Body	base_Class.generalization -> forAll(general.oclIsTypeOf(BusinessComponent) and specific.oclIsTypeOf(BusinessComponent))

— PropertiesMustBeBusinessElements

Description

The Properties of a <<BusinessComponent>>-stereotyped Class shall be stereotyped by a concrete subclass of <<BusinessElement>>.

Context	ISO20022::Profile::ConceptualLevel::Static::BusinessComponent
Language	OCL2.0
Body	base_Class.ownedAttribute -> forAll (attribute |attribute.oclIsTypeOf(BusinessElement))

A.2.5.3 Stereotype BusinessElement

Description

Qualified name	ISO20022::Profile::ConceptualLevel::Static::BusinessElement
Abstract	false
Owner	ISO20022::Profile::ConceptualLevel::Static
Base UML Metaclass	Property
Superclass	ISO20022::Profile::RepositoryConcept

Constraints

— AssociationEndsMustBeNavigable

Description

A <<BusinessElement>>-stereotyped association end shall be navigable and the opposite end shall also be stereotyped <<BusinessElement>>.

Context	ISO20022::Profile::ConceptualLevel::Static::BusinessElement
Language	OCL2.0
Body	not(base_Property.association.oclIsUndefined()) implies (base_Property.association.navigableOwnedEnd -> includes(base_Property) and base_Property.opposite.oclIsTypeOf(BusinessElement))

— OwnerIsABusinessComponent

Description

A <<BusinessElement>>-stereotyped Property shall be a Property of a <<BusinessComponent>>-stereotyped Class.

Context	ISO20022::Profile::ConceptualLevel::Static::BusinessElement
Language	OCL2.0
Body	base_Property.owner.oclIsTypeOf(BusinessComponent)

— SharedAggregationNotAllowed

Description

A BusinessAssociation may only have NONE or COMPOSITE aggregation, i.e. the aggregation on a <<BusinessElement>>-stereotyped Property may not be SHARED.

Context	ISO20022::Profile::ConceptualLevel::Static::BusinessElement
Language	OCL2.0
Body	not(base_Property.aggregation = AggregationKind::SHARED)

— TypeOfAnAssociationEndIsABusinessComponent

Description

The type of a <<BusinessElement>>-stereotyped association end shall be a <<BusinessComponent>>-stereotyped Class.

Context	ISO20022::Profile::ConceptualLevel::Static::BusinessElement
Language	OCL2.0
Body	not(base_Property.association.oclIsUndefined()) implies base_Property.type.oclIsTypeOf(BusinessComponent)

A.2.6 Package ISO20022::Profile::ConceptualToLogicalTransformation

A.2.6.1 General

Applied Stereotype	<<profile>>

Description

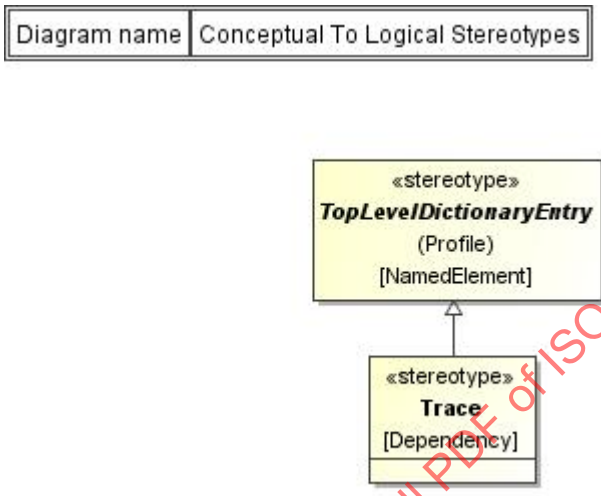

Figure A.5 — Example of Conceptual To Logical Stereotypes

A.2.6.2 Stereotype Trace

Description

Qualified name	ISO20022::Profile::ConceptualToLogicalTransformation::Trace
Abstract	false
Owner	ISO20022::Profile::ConceptualToLogicalTransformation
Base UML Metaclass	Dependency
Superclass	ISO20022::Profile::TopLevelDictionaryEntry

Constraints

— CardinalityAlignment

Description

A <<Trace>> from a MessageElement to a BusinessElement can only raise the minimum cardinality and can only lower the maximum cardinality.

Context	ISO20022::Profile::ConceptualToLogicalTransformation::Trace
Language	OCL2.0

Body	-- From UML Superstructure spec, MultiplicityElement has -- a query (includesMultiplicity) to check cardinality compliance base_Dependency.client -> forAll(cl : NamedElement| (cl.oclIsKindOf(MultiplicityElement) and cl.oclIsTypeOf(MessageElement)) implies base_Dependency.supplier -> forAll(sup : NamedElement | (sup.oclIsKindOf(MultiplicityElement) and sup.oclIsTypeOf(BusinessElement)) implies (sup.oclAsType(MultiplicityElement). includesMultiplicity(cl.oclAsType(MultiplicityElement)))))

— LegalTraces

Description

The following <<Trace>>-stereotyped Dependencies are the only ones that are legal:
- A <<MessageComponentType>> dependent on a <<BusinessComponent>>
- A <<MessageElement>> dependent on a <<BusinessElement>>
- A <<MessageElement>> dependent on a <<BusinessComponent>>
- A <<CodeSet>> dependent on a <<CodeSet>>

Context	ISO20022::Profile::ConceptualToLogicalTransformation::Trace
Language	OCL2.0
Body	(base_Dependency.client.oclIsTypeOf(MessageComponent) and base_Dependency.supplier.oclIsTypeOf(BusinessComponent)) or (base_Dependency.client.oclIsTypeOf(ChoiceComponent) and base_Dependency.supplier.oclIsTypeOf(BusinessComponent)) or (base_Dependency.client.oclIsTypeOf(MessageElement) and base_Dependency.supplier.oclIsTypeOf(BusinessElement)) or (base_Dependency.client.oclIsTypeOf(MessageElement) and base_Dependency.supplier.oclIsTypeOf(BusinessComponent)) or (base_Dependency.client.oclIsTypeOf(CodeSet) and base_Dependency.supplier.oclIsTypeOf(CodeSet))

A.2.7 Package ISO20022::Profile::DataTypes

A.2.7.1 General

Applied Stereotype	<<profile>>

Description

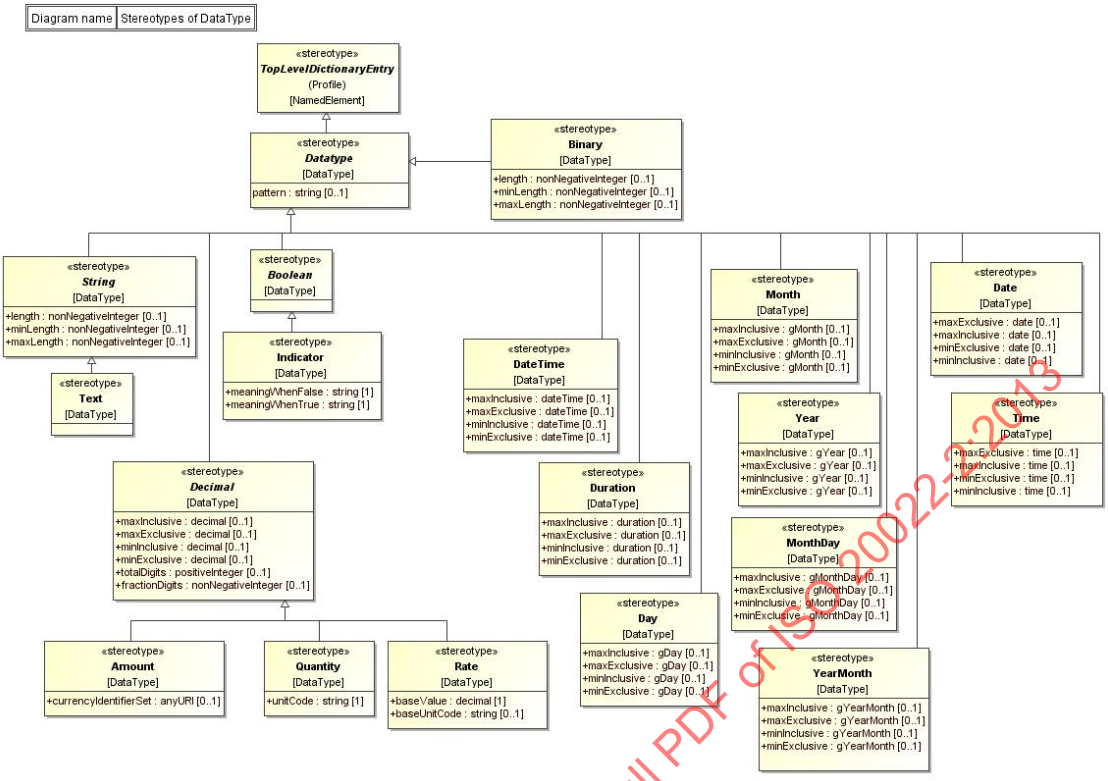

Figure A.6 — Example of stereotypes of DataType

A.2.7.2 Stereotype Amount

Description

Qualified name	ISO20022::Profile::DataTypes::Amount
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Decimal

Tag Definitions

— currencyIdentifierSet

Type	ISO20022::TypeLibrary::XMLSchema::anyURI
Owner	ISO20022::Profile::DataTypes::Amount
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.3 Stereotype Binary

Description

Qualified name	ISO20022::Profile::DataTypes::Binary
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— length

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::DataTypes::Binary
Default Value	
Multiplicity	0..1
Ordered	false

— maxLength

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::DataTypes::Binary
Default Value	
Multiplicity	0..1
Ordered	false

— minLength

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::DataTypes::Binary
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.4 Stereotype Boolean

Description

Qualified name	ISO20022::Profile::DataTypes::Boolean
Abstract	true

Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

A.2.7.5 Stereotype Datatype

Description

Qualified name	ISO20022::Profile::DataTypes::Datatype
Abstract	true
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::TopLevelDictionaryEntry

Tag Definitions

— pattern

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::DataTypes::Datatype
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.6 Stereotype Date

Description

Qualified name	ISO20022::Profile::DataTypes::Date
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— maxExclusive

Type	ISO20022::TypeLibrary::XMLSchema::date
Owner	ISO20022::Profile::DataTypes::Date
Default Value	
Multiplicity	0..1
Ordered	false

— maxInclusive

Type	ISO20022::TypeLibrary::XMLSchema::date
Owner	ISO20022::Profile::DataTypes::Date
Default Value	
Multiplicity	0..1
Ordered	false

— minExclusive

Type	ISO20022::TypeLibrary::XMLSchema::date
Owner	ISO20022::Profile::DataTypes::Date
Default Value	
Multiplicity	0..1
Ordered	false

— minInclusive

Type	ISO20022::TypeLibrary::XMLSchema::date
Owner	ISO20022::Profile::DataTypes::Date
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.7 Stereotype DateTime

Description

Qualified name	ISO20022::Profile::DataTypes::DateTime
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— maxExclusive

Type	ISO20022::TypeLibrary::XMLSchema::dateTime
Owner	ISO20022::Profile::DataTypes::DateTime

Default Value	
Multiplicity	0..1
Ordered	false

— maxInclusive

Type	ISO20022::TypeLibrary::XMLSchema::dateTime
Owner	ISO20022::Profile::DataTypes::DateTime
Default Value	
Multiplicity	0..1
Ordered	false

— minExclusive

Type	ISO20022::TypeLibrary::XMLSchema::dateTime
Owner	ISO20022::Profile::DataTypes::DateTime
Default Value	
Multiplicity	0..1
Ordered	false

— minInclusive

Type	ISO20022::TypeLibrary::XMLSchema::dateTime
Owner	ISO20022::Profile::DataTypes::DateTime
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.8 Stereotype Day

Description

Qualified name	ISO20022::Profile::DataTypes::Day
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— maxExclusive

Type	ISO20022::TypeLibrary::XMLSchema::gDay
Owner	ISO20022::Profile::DataTypes::Day
Default Value	
Multiplicity	0..1
Ordered	false

— maxInclusive

Type	ISO20022::TypeLibrary::XMLSchema::gDay
Owner	ISO20022::Profile::DataTypes::Day
Default Value	
Multiplicity	0..1
Ordered	false

— minExclusive

Type	ISO20022::TypeLibrary::XMLSchema::gDay
Owner	ISO20022::Profile::DataTypes::Day
Default Value	
Multiplicity	0..1
Ordered	false

— minInclusive

Type	ISO20022::TypeLibrary::XMLSchema::gDay
Owner	ISO20022::Profile::DataTypes::Day
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.9 Stereotype Decimal

Description

Qualified name	ISO20022::Profile::DataTypes::Decimal
Abstract	true
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— fractionDigits

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::DataTypes::Decimal
Default Value	
Multiplicity	0..1
Ordered	false

— maxExclusive

Type	ISO20022::TypeLibrary::XMLSchema::decimal
Owner	ISO20022::Profile::DataTypes::Decimal
Default Value	
Multiplicity	0..1
Ordered	false

— maxInclusive

Type	ISO20022::TypeLibrary::XMLSchema::decimal
Owner	ISO20022::Profile::DataTypes::Decimal
Default Value	
Multiplicity	0..1
Ordered	false

— minExclusive

Type	ISO20022::TypeLibrary::XMLSchema::decimal
Owner	ISO20022::Profile::DataTypes::Decimal
Default Value	

Multiplicity	0..1
Ordered	false

— minInclusive

Type	ISO20022::TypeLibrary::XMLSchema::decimal
Owner	ISO20022::Profile::DataTypes::Decimal
Default Value	
Multiplicity	0..1
Ordered	false

— totalDigits

Type	ISO20022::TypeLibrary::XMLSchema::positiveInteger
Owner	ISO20022::Profile::DataTypes::Decimal
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.10 Stereotype Duration

Description

Qualified name	ISO20022::Profile::DataTypes::Duration
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— maxExclusive

Type	ISO20022::TypeLibrary::XMLSchema::duration
Owner	ISO20022::Profile::DataTypes::Duration
Default Value	
Multiplicity	0..1
Ordered	false

— maxInclusive

Type	ISO20022::TypeLibrary::XMLSchema::duration
Owner	ISO20022::Profile::DataTypes::Duration
Default Value	
Multiplicity	0..1
Ordered	false

— minExclusive

Type	ISO20022::TypeLibrary::XMLSchema::duration
Owner	ISO20022::Profile::DataTypes::Duration
Default Value	
Multiplicity	0..1
Ordered	false

— minInclusive

Type	ISO20022::TypeLibrary::XMLSchema::duration
Owner	ISO20022::Profile::DataTypes::Duration
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.11 Stereotype Indicator

Description

Qualified name	ISO20022::Profile::DataTypes::Indicator
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Boolean

Tag Definitions

— meaningWhenFalse

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::DataTypes::Indicator

Default Value	
Multiplicity	1
Ordered	false

— meaningWhenTrue

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::DataTypes::Indicator
Default Value	
Multiplicity	1
Ordered	false

A.2.7.12 Stereotype Month

Description

Qualified name	ISO20022::Profile::DataTypes::Month
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— maxExclusive

Type	ISO20022::TypeLibrary::XMLSchema::gMonth
Owner	ISO20022::Profile::DataTypes::Month
Default Value	
Multiplicity	0..1
Ordered	false

— maxInclusive

Type	ISO20022::TypeLibrary::XMLSchema::gMonth
Owner	ISO20022::Profile::DataTypes::Month
Default Value	
Multiplicity	0..1
Ordered	false

— minExclusive

Type	ISO20022::TypeLibrary::XMLSchema::gMonth
Owner	ISO20022::Profile::DataTypes::Month
Default Value	
Multiplicity	0..1
Ordered	false

— minInclusive

Type	ISO20022::TypeLibrary::XMLSchema::gMonth
Owner	ISO20022::Profile::DataTypes::Month
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.13 Stereotype MonthDay

Description

Qualified name	ISO20022::Profile::DataTypes::MonthDay
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— maxExclusive

Type	ISO20022::TypeLibrary::XMLSchema::gMonthDay
Owner	ISO20022::Profile::DataTypes::MonthDay
Default Value	
Multiplicity	0..1
Ordered	false

— maxInclusive

Type	ISO20022::TypeLibrary::XMLSchema::gMonthDay
Owner	ISO20022::Profile::DataTypes::MonthDay

Default Value	
Multiplicity	0..1
Ordered	false

— minExclusive

Type	ISO20022::TypeLibrary::XMLSchema::gMonthDay
Owner	ISO20022::Profile::DataTypes::MonthDay
Default Value	
Multiplicity	0..1
Ordered	false

— minInclusive

Type	ISO20022::TypeLibrary::XMLSchema::gMonthDay
Owner	ISO20022::Profile::DataTypes::MonthDay
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.14 Stereotype Quantity

Description

Qualified name	ISO20022::Profile::DataTypes::Quantity
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Decimal

Tag Definitions

— unitCode

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::DataTypes::Quantity
Default Value	
Multiplicity	1
Ordered	false

A.2.7.15 Stereotype Rate

Description

Qualified name	ISO20022::Profile::DataTypes::Rate
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Decimal

Tag Definitions

— baseUnitCode

Type	ISO20022::TypeLibrary::XMLSchema::string
Owner	ISO20022::Profile::DataTypes::Rate
Default Value	
Multiplicity	0..1
Ordered	false

— baseValue

Type	ISO20022::TypeLibrary::XMLSchema::decimal
Owner	ISO20022::Profile::DataTypes::Rate
Default Value	
Multiplicity	1
Ordered	false

A.2.7.16 Stereotype String

Description

Qualified name	ISO20022::Profile::DataTypes::String
Abstract	true
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions

— length

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::DataTypes::String

Default Value	
Multiplicity	0..1
Ordered	false

— maxLength

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::DataTypes::String
Default Value	
Multiplicity	0..1
Ordered	false

— minLength

Type	ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger
Owner	ISO20022::Profile::DataTypes::String
Default Value	
Multiplicity	0..1
Ordered	false

A.2.7.17 Stereotype Text

Description

Qualified name	ISO20022::Profile::DataTypes::Text
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::String

A.2.7.18 Stereotype Time

Description

Qualified name	ISO20022::Profile::DataTypes::Time
Abstract	false
Owner	ISO20022::Profile::DataTypes
Base UML Metaclass	DataType
Superclass	ISO20022::Profile::DataTypes::Datatype

Tag Definitions