BS ISO 20022-1:2013

BSI Standards Publication

# Financial services — Universal financial industry message scheme

## Part 1: Metamodel

bsi.

...making excellence a habit.™

**National foreword**

This British Standard is the UK implementation of ISO 20022-1:2013. It supersedes BS ISO 20022-1:2004 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/12, Financial services.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.



ISBN 978 0 580 70956 2

ICS 03.060

**Compliance with a British Standard cannot confer immunity from legal obligations.**

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2013.

**Amendments issued since publication**

| Date | Text affected |
|---|---|

BS ISO 20022-1:2013

INTERNATIONAL STANDARD

**ISO 20022-1**

Second edition
2013-05-01

# Financial services — Universal financial industry message scheme —

## Part 1: **Metamodel**

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

*Partie 1: Métamodèle*

Reference number
ISO 20022-1:2013(E)

## COPYRIGHT PROTECTED DOCUMENT



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 20022-1 was prepared by Technical Committee ISO/TC 68, *Financial services*.

This second edition cancels and replaces the first edition (ISO 20022-1:2004), 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 hereon 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 this part of ISO 20022, 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.

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

— ISO 20022-2 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.

**INTERNATIONAL STANDARD** **ISO 20022-1:2013(E)**

# Financial services — Universal financial industry message scheme —

# Part 1: **Metamodel**

## 1 Scope

This part of ISO 20022 consists of:

— the overall description of the modelling approach;

— the overall description of the ISO 20022 Repository contents;

— a high-level description of the input to be accepted by the Registration Authority to feed/modify the Repository's DataDictionary and BusinessProcessCatalogue;

— a high-level description of the Repository output to be made publicly available by the Registration Authority.

BusinessTransactions and Message Sets complying with ISO 20022 can be used for electronic data interchange amongst any industry participants (financial and others), independently of any specific communication network. Network-dependent rules, such as message acknowledgement and message protection, are outside the scope of ISO 20022.

## 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 20022-2, *Financial services — Universal financial industry message scheme — UML profile*

The Zachman Framework for Enterprise Architecture — Zachman Institute for Framework Advancement

W3C Recommendation: XML Schema Part 2: Datatypes Second Edition, 28 October 2004

## 3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

**3.1**
**Address**
identification and efficient resolution to the location of a MessagingEndpoint

NOTE The purpose of an Address is to efficiently resolve a location. This is what distinguishes an Address from any other Identifier, which merely identifies something.

**3.2**
**amount**
number of monetary units specified in a currency where the unit of currency is explicit or implied

**3.3**
**binary**
any set of values drawn from the value space of 'base64Binary', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.4**
**boolean**
any set of values drawn from the value space of 'boolean', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.5**
**BroadcastList**
set of references to MessagingEndpoints (identified by their Address), that is used for message broadcasting.

NOTE 1 The BroadcastList is managed by the MessageTransportSystem, which provides a mechanism to maintain the BroadcastList.

NOTE 2 “Set” means the list of Addresses is unordered and each Address may only be present once.

**3.6**
**BusinessArea**
set of strongly related business activities that provide a self-standing business value to a set of BusinessRoles

EXAMPLE Securities pre-trade, payment initiation.

NOTE BusinessAreas are stored in the BusinessProcessCatalogue.

**3.7**
**BusinessAssociation**
relation between two BusinessComponents

EXAMPLE The servicing of an account by a party.

NOTE 1 BusinessAssociations are a category of BusinessConcepts. Their meaning can only be described unambiguously in combination with these two BusinessComponents.

NOTE 2 There can be several BusinessAssociations between two particular BusinessComponents.

**3.8**
**BusinessAssociationEnd**
the endpoint of a BusinessAssociation, which connects the BusinessAssociation to the BusinessComponent

**3.9**
**BusinessAttribute**
a BusinessElement, typed by a BusinessComponent or a DataType (contrary to a BusinessAssociationEnd, which is always typed by another BusinessComponent)

EXAMPLE AccountIdentification, PhoneNumber.

**3.10**
**BusinessComponent**
representation of a (part of a) key business notion, characterized by specific BusinessElements

EXAMPLE Account, trade, party.

NOTE 1 BusinessComponents are a category of BusinessConcepts. They are stored in the DataDictionary.

NOTE 2 A BusinessComponent can have one or more semantic relations with other BusinessComponents.

**3.11**
**BusinessComponentTrace**
semantic relationship between a MessageComponentType/MessageElement and the BusinessComponent from which it is derived

**3.12**
**BusinessConcept**
a DataDictionary item defined at the Conceptual level with a business meaning

**3.13**
**BusinessElement**
property of a BusinessComponent that has a distinctive meaning within the scope of that BusinessComponent

EXAMPLE Account status, deal price, trade date and deal time.

**3.14**
**BusinessElementTrace**
semantic relationship between a MessageElement and the BusinessElement from which it is derived

**3.15**
**BusinessProcess**
unrealized definition of the business activities undertaken by BusinessRoles within a BusinessArea whereby each BusinessProcess fulfils one type of business activity and whereby a BusinessProcess might include and extend other BusinessProcesses

EXAMPLE Securities ordering, trade matching.

NOTE 1 A BusinessProcess can contain other BusinessProcesses such as in a hierarchical structure.

NOTE 2 BusinessProcesses are stored in the BusinessProcessCatalogue.

**3.16**
**BusinessProcessCatalogue**
that part of the ISO 20022 Repository which contains all items related to Business Process and BusinessTransaction

NOTE It contains related items from the BusinessArea down to the MessageDefinitions and their physical implementation.

**3.17**
**BusinessProcessTrace**
relationship between a BusinessTransaction and the BusinessProcess on which the BusinessTransaction is based

**3.18**
**BusinessRole**
functional role played by a business actor in a particular BusinessProcess or BusinessTransaction

EXAMPLE Account owner, buyer.

NOTE 1 BusinessRoles are a category of BusinessConcepts and are stored in the DataDictionary.

NOTE 2 A business actor can play different BusinessRoles in different BusinessProcesses.

**3.19**
**BusinessRoleTrace**
relationship between a Participant in a BusinessTransaction and a BusinessRole identified in the BusinessProcess from which the BusinessTransaction is derived

**3.20**
**BusinessTransaction**
particular solution that meets the communication requirements and the interaction requirements of a particular BusinessProcess and BusinessArea

NOTE It is typically based on the use of MessageInstances.

**3.21**
**BusinessTransactionTrace**
relationship between a BusinessTransaction and its physical implementation

**3.22**
**ChoiceComponent**
re-usable Dictionary Item that is a building block for assembling MessageDefinitions, composed of a choice of MessageElements

NOTE 1 It is usually linked to a BusinessComponent.

NOTE 2 ChoiceComponents are stored in the DataDictionary.

**3.23**
**Code**
character string (letters, figures or symbols) that for brevity and/or language independence can be used to represent or replace a definitive value or text of an attribute

**3.24**
**CodeSet**
complete and enumerated set of Codes grouped together to characterize all the values of an attribute

**3.25**
**CodeSetTrace**
semantic relationship between two CodeSets whereby the derived Codeset is used as the type of a BusinessElement and the deriving Codeset is used as the type of a MessageElement

**3.26**
**Constraint**
rule that shall be universally satisfied, i.e. all conditions required for the Constraint to be applicable are known

EXAMPLE An Account has an AccountOwner.

**3.27**
**ConvergenceDocumentation**
documentation set showing relations between ISO 20022 MessageDefinitions, MessageComponentTypes, MessageElements, BusinessComponents and/or BusinessElements and items defined in other industry MessageSets

**3.28**
**Conversation**
exchange of one or more MessageInstances amongst MessagingEndpoints

NOTE In a synchronous Conversation, the sending MessagingEndpoint blocks the sending and receipt of other TransportMessages within the conversation, in which the TransportMessage was sent, while waiting for a response to this sent TransportMessage. This is not the case in an asynchronous conversation.

**3.29**
**DataDictionary**
part of the ISO 20022 Repository that contains all items that can be re-used during business process modelling and message definition activities

NOTE The DataDictionary therefore contains BusinessConcepts, MessageConcepts and DataTypes.

**3.30**
**DataType**
representation of a set of values without identity

**3.31**
**date**
any set of values drawn from the value space of 'date', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.32**
**dateTime**
any set of values drawn from the value space of 'dateTime', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.33**
**day**
any set of values drawn from the value space of 'gDay', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.34**
**decimal**
any set of values drawn from the value space of 'decimal', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.35**
**duration**
any set of values drawn from the value space of 'duration', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.36**
**ExternalSchema**
reusable Dictionary Item that allows referral to a structure defined outside the ISO 20022 MessageDefinition

EXAMPLE In case of XML (eXtensible Markup Language), this artefact is transformed into an XML Schema "any" element and the external structure is defined through another XML Schema.

**3.37**
**IdentifierSet**
unenumerated set of values outside the Repository whereby each value distinguishes uniquely one instance of an object within an identification scheme from all other instances of the objects within the same scheme

**3.38**
**indicator**
a list of exactly two mutually exclusive values that express the only two possible states of an instance of an object

**3.39**
**industryMessageSet**
set of non-ISO 20022 compliant messages, which is defined and used by part of the financial industry

EXAMPLE The set of FIX v5 messages.

**3.40**
**ISO15022MessageSet**
industryMessageSet constructed according to the rules defined in ISO 15022-1 and ISO 15022-2, which is stored in the ISO 15022 Catalogue of Messages

**3.41**
**MessageAssociationEnd**
type of MessageElement that specifies the role of a MessageAssociation

**3.42**
**MessageAttribute**
type of MessageElement which is either a DataType or a MessageComponentType

**3.43**
**MessageBuildingBlock**
characteristic of a MessageDefinition that has a unique meaning within the scope of that MessageDefinition

NOTE MessageBuildingBlocks are not reused, since they only have meaning within a specific MessageDefinition.

**3.44**
**MessageChoreography**
precise and complete description of a MessageInstance exchange within a BusinessTransaction, describing the sequence and correlation of MessageInstances within a conversation, including the constraints on the interaction between Participants

NOTE Every BusinessTransaction contains its own MessageChoreography.

**3.45**
**MessageComponent**
re-usable Dictionary Item that is a building block for assembling MessageDefinitions, composed of a sequence of MessageElements

EXAMPLE Trade Details, which contains a number of the properties of the related BusinessComponent “Trade”.

**3.46**
**MessageComponentType**
MessageComponent, ExternalSchema or ChoiceComponent

NOTE 1 When a MessageComponentType has a business meaning it is linked to a BusinessComponent.

NOTE 2 MessageComponentTypes are a category of MessageConcepts and are stored in the DataDictionary.

**3.47**
**MessageConcept**
DataDictionary artefact, which is not a DataType, that is used in a MessageDefinition

**3.48**
**MessageDefinition**
formal description of the structure of a MessageInstance

NOTE 1 The MessageDefinition is built as a tree structure of MessageComponentTypes and DataTypes. A MessageDefinition is uniquely identified in the BusinessProcessCatalogue.

NOTE 2 A MessageDefinition can have several market practices.

**3.49**
**MessageDefinitionIdentifier**
unique identification of a MessageDefinition within the ISO 20022 namespace, identifying the BusinessArea to which the MessageDefinition belongs, the Message Functionality it covers, its flavour and its version

**3.50**
**MessageDefinitionTrace**
relationship between a MessageDefinition and its physical implementation as a SyntaxMessageScheme

NOTE This relationship is explained in ISO 20022-4.

**3.51**
**MessageElement**
characteristic of a MessageComponent/ChoiceComponent, which has a unique meaning within the scope of that MessageComponent/ChoiceComponent

EXAMPLE Trade Date and Time, as part of the MessageComponent “Trade Details”.

NOTE MessageElements are a category of MessageConcepts. They are stored in the DataDictionary where they are owned by a particular MessageComponent/ChoiceComponent. Their meaning can only be described unambiguously in combination with that MessageComponent/ChoiceComponent.

**3.52**
**MessageInstance**
instance of MessageDefinition, containing a set of structured information exchanged between BusinessRoles, in the scope of a BusinessTransaction

EXAMPLE Notice Of Execution, Order To Buy, Credit Transfer.

NOTE A MessageInstance is expected to be valid against the related MessageDefinition in the ISO 20022 Repository. This implies validity against the SyntaxMessageScheme as well as validity against the Constraints and market practices that are registered for this MessageDefinition.

**3.53**
**MessageSet**
set of MessageDefinitions

NOTE MessageDefinitions within a MessageSet do not have to belong to the same BusinessArea.

**3.54**
**MessageTransmission**
passing of information from one Participant to another in the context of a BusinessTransaction

**3.55**
**MessageTransportMode**
group of settings for the values for the MessageTransportCharacteristics properties

NOTE 1 A MessageTransportMode is named and registered in the ISO 20022 Repository. Each MessageTransportCharacteristic is given a value.

NOTE 2 A MessageTransportMode can be associated with many BusinessTransactions. The MessageTransportMode is used to organize commonly used combinations of MessageTransportCharacteristic settings.

**3.56**
**MessageTransportSystem**
mechanism that receives Transport Messages from the sending MessagingEndpoint, transports them, and delivers them to the receiving MessagingEndpoint

NOTE 1 The MessageTransportSystem is responsible for delivering Transport Messages to each Addressee.

NOTE 2 The purpose of the MessageTransportSystem is to provide a clear delineation of the responsibility of the MessagingEndpoints and any MessageTransportSystem service providers. The role can be fulfilled by the sending MessagingEndpoint or by a separate service provider who provides a MessageTransportSystem. MessagingTransportSystems can be chained together into a single MessageTransportSystem

**3.57**
**MessageTypeTrace**
relationship between a MessageTransmission in a BusinessTransaction and its corresponding MessageDefinition

**3.58**
**MessagingEndpoint**
addressable node on the MessageTransportSystem which is capable of sending and receiving TransportMessages

NOTE A MessagingEndpoint has an Address.

**3.59**
**month**
any set of values drawn from the value space of 'gMonth', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.60**
**MonthDay**
any set of values drawn from the value space of 'gMonthDay', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.61**
**Participant**
involvement of a BusinessRole in a BusinessTransaction

**3.62**
**quantity**
a counted number of non-monetary units possibly including fractions

**3.63**
**rate**
a quantity or amount measured with respect to another measured quantity or amount

EXAMPLE US Dollars per hour, US Dollars per EURO.

**3.64**
**receive**
handling of a stimulus passed from a sender instance

**3.65**
**Repository**
place where all RepositoryConcepts are stored

**3.66**
**RepositoryConcept**
artefact that has been defined during the development of an ISO 20022 MessageDefinition and which is stored in the Repository

**3.67**
**send**
passing of a stimulus from a sender instance to a receiver instance

**3.68**
**string**
any set of values drawn from the value space of 'string', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.69**
**SyntaxMessageScheme**
syntax processable notation used to define the structure of a MessageInstance in a particular syntax

NOTE 1 In case of XML, the representation might, for instance, be an XML DTD or an XML Schema and can then be used as a validation tool for MessageInstances.

NOTE 2 Syntax message representations are stored in the BusinessProcessCatalogue

**3.70**
**text**
finite set of characters

**3.71**
**time**
any set of values drawn from the value space of 'time', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.72**
**TopLevelCatalogueEntry**
artefact in the BusinessProcessCatalogue that is not owned by another artefact in the Repository

**3.73**
**TopLevelDictionaryEntry**
artefact in the Dictionary that is not owned by another artefact in the Repository

**3.74**
**trace**
relationship between two objects that represent the same concept but have a different but related context

**3.75**
**TransportMessage**
document that is an instance of the MessageTransportSystem message schema

**3.76**
**year**
any set of values drawn from the value space of 'gYear', as specified by W3C Recommendation XML Schema Part 2: Datatypes

**3.77**
**yearMonth**
any set of values drawn from the value space of 'gYearMonth', as specified by W3C Recommendation XML Schema Part 2: Datatypes

## 4 Type Library

The Type Library contains the primitive data types used in both this International Standard's metamodel and the models created in accordance with this International Standard. It consists of XSD (XML Schema Definition) Built-in DataTypes and Enumerations. Further details on these packages can be found in Annex A.

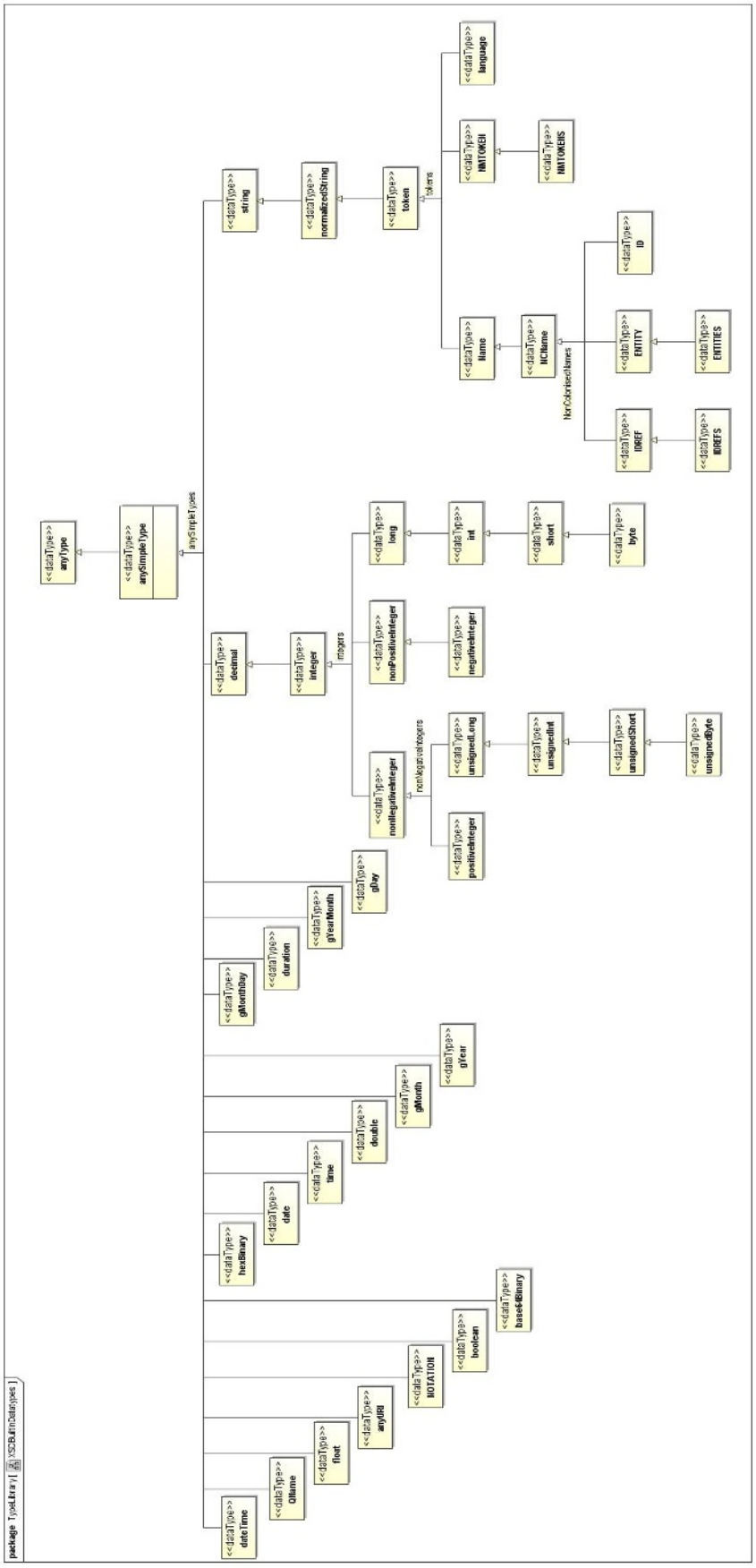


**Figure 1 — XSD Built-in DataTypes**

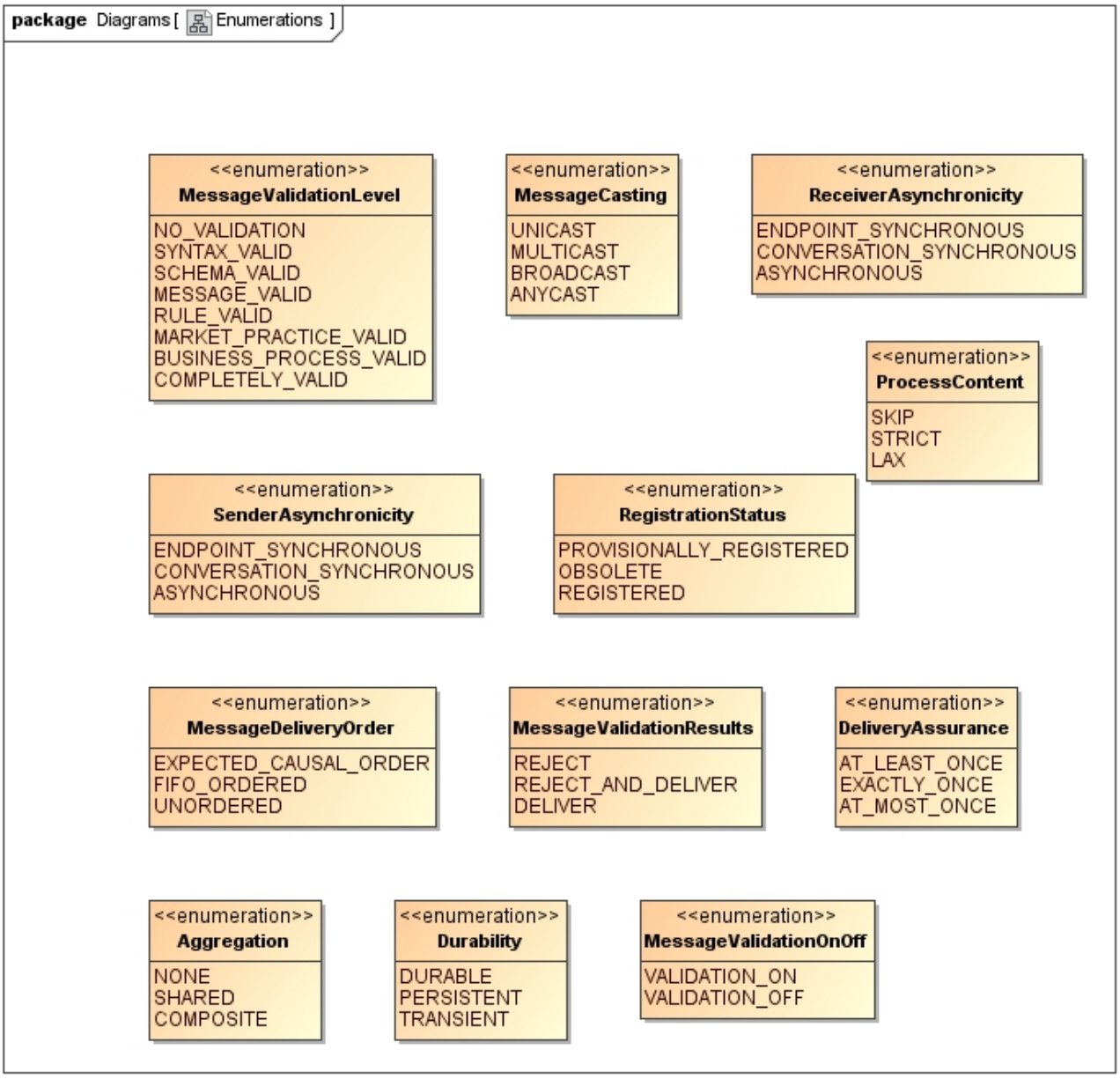


**Figure 2 — DataType enumerated values**

# 5 Metamodel packages

## 5.1 General

The metamodel describes the structure of models built in accordance with this International Standard. All models produced according to this International Standard shall comply with this one model.

## 5.2 The metamodel's use of ISO20022::TypeLibrary

The metamodel imports the ISO20022::TypeLibrary Package and its subpackages, which are defined in Annex A. It uses the types defined therein to define the metamodel.

NOTE The ISO20022::TypeLibrary Package is not contained within the ISO20022::Metamodel Package because it is used by both the metamodel and the UML Profile defined in ISO 20022-2, which also imports it. The ISO20022::TypeLibrary::XMLSchema Package 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.

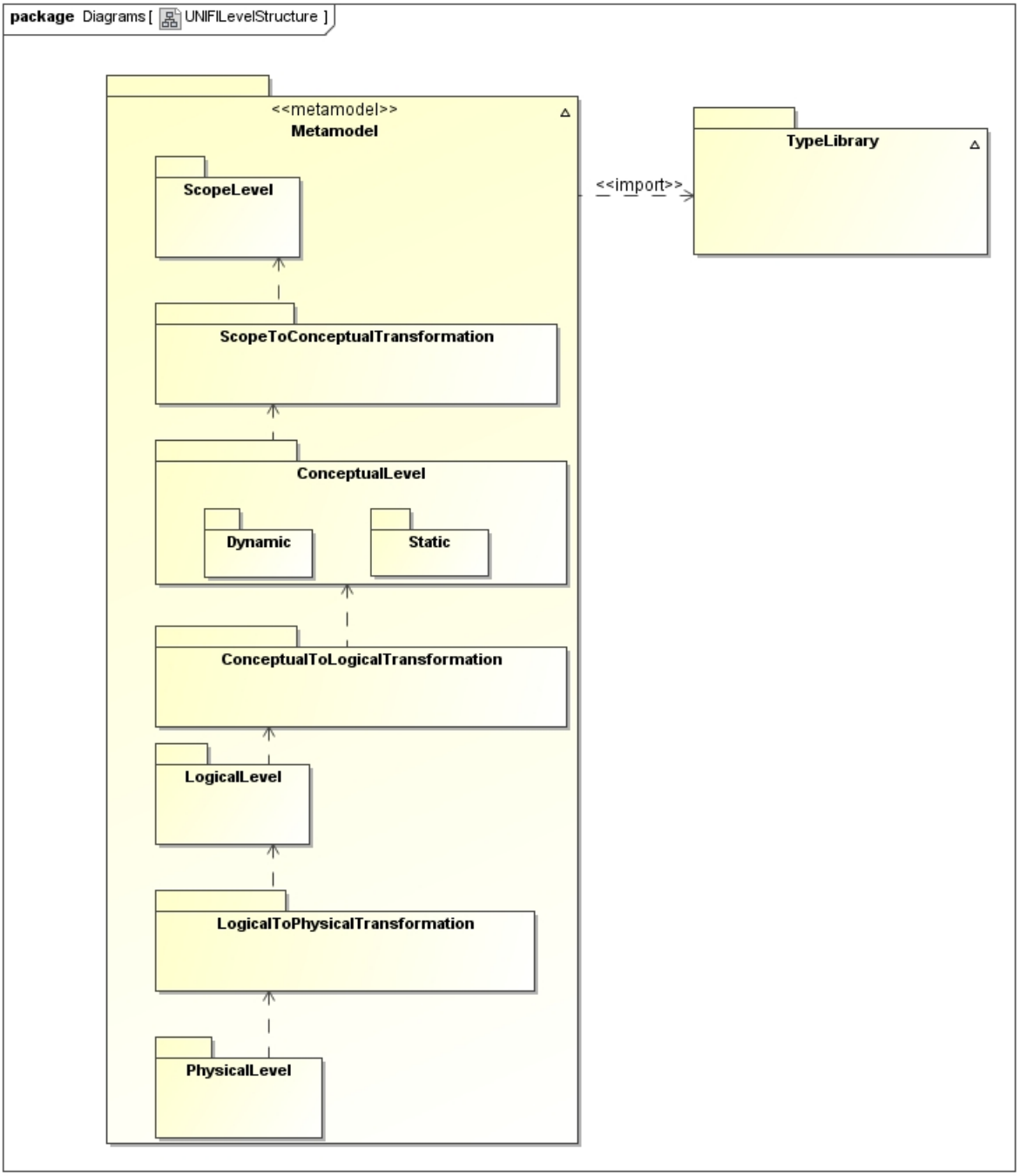


**Figure 3 — ISO 20022 level structure**

## 5.3 Levels

The metamodel has four levels of model, each of increasing realization. These four levels are based upon the first four levels of the Zachman Framework.

**Table 1 — Metamodel levels**

| Level name | Focus |
|---|---|
| Scope | Achieving a thorough understanding of the business objectives of the considered BusinessArea and its relevant Business Processes. |
| Conceptual | Formalizing the semantics and discovering the communication and interaction requirements related to these Business Processes by defining the BusinessTransactions, BusinessActivities and Message Choreographies related to these Business Processes. |
| Logical | Creating a precise description of the messages and systems, without regard to technology. |
| Physical | Creating a precise description of the messages and systems in a technology that can be used for implementation. |

The Zachmann Framework has two further levels. These are regarded as being under the aegis of the implementers of this International Standard.

For a BusinessArea being modelled, each level is modelled completely and precisely.

The number of levels may not be added to or reduced.

NOTE 1 Adding or removing levels introduces redundancy or discontinuity.

Each level is disjointed.

NOTE 2 No concept may appear at more than one level without a change in the level of realization.

There is no relationship between the levels other than realization.

There is no sequence, other than the ordering of realization, implicit in the levels. Any level may be completed in any order.

NOTE 3 This might be further constrained in a modelling method.

NOTE 4 The levels do not subsume each other.

NOTE 5 The metaclasses used to explain the different levels are detailed in Annex B.

### 5.3.1 Scope level

The purpose of the Scope level is to acquire an understanding of the BusinessArea for which an ISO 20022 compliant BusinessTransaction and MessageSet is to be developed. Describing the BusinessProcesses helps in the identification of the communication problems that exist among the business actors (modelled as BusinessRoles) that take part in these processes. Those communication problems will be the main drivers for the Conceptual level. Identifying the Business Information that is manipulated in a BusinessArea is also important for the later Logical level, because the MessageDefinitions will contain data elements that are related to this business information.

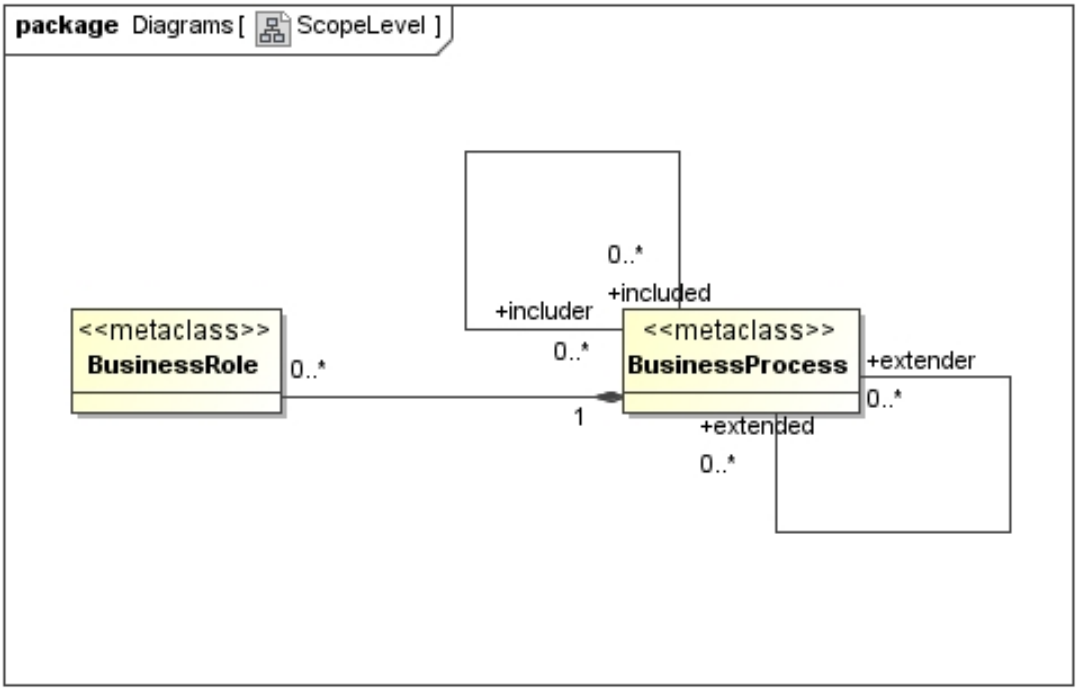


**Figure 4 — Scope level**

### 5.3.2 Conceptual level

#### 5.3.2.1 Creation of a semantic model of all artefacts (static part)

The Conceptual level starts by using a “black box perspective” in order to avoid prematurely tackling architectural issues regarding the issues encountered during the implementation. A “black box” perspective means that one does not try to define BusinessTransactions and MessageDefinitions. The focus is only on defining “who needs what” in order to execute the BusinessProcesses. There is no attempt yet to define how to get the information at the right moment to the right business user. All BusinessConcepts that are involved in the BusinessTransactions shall firstly be identified.

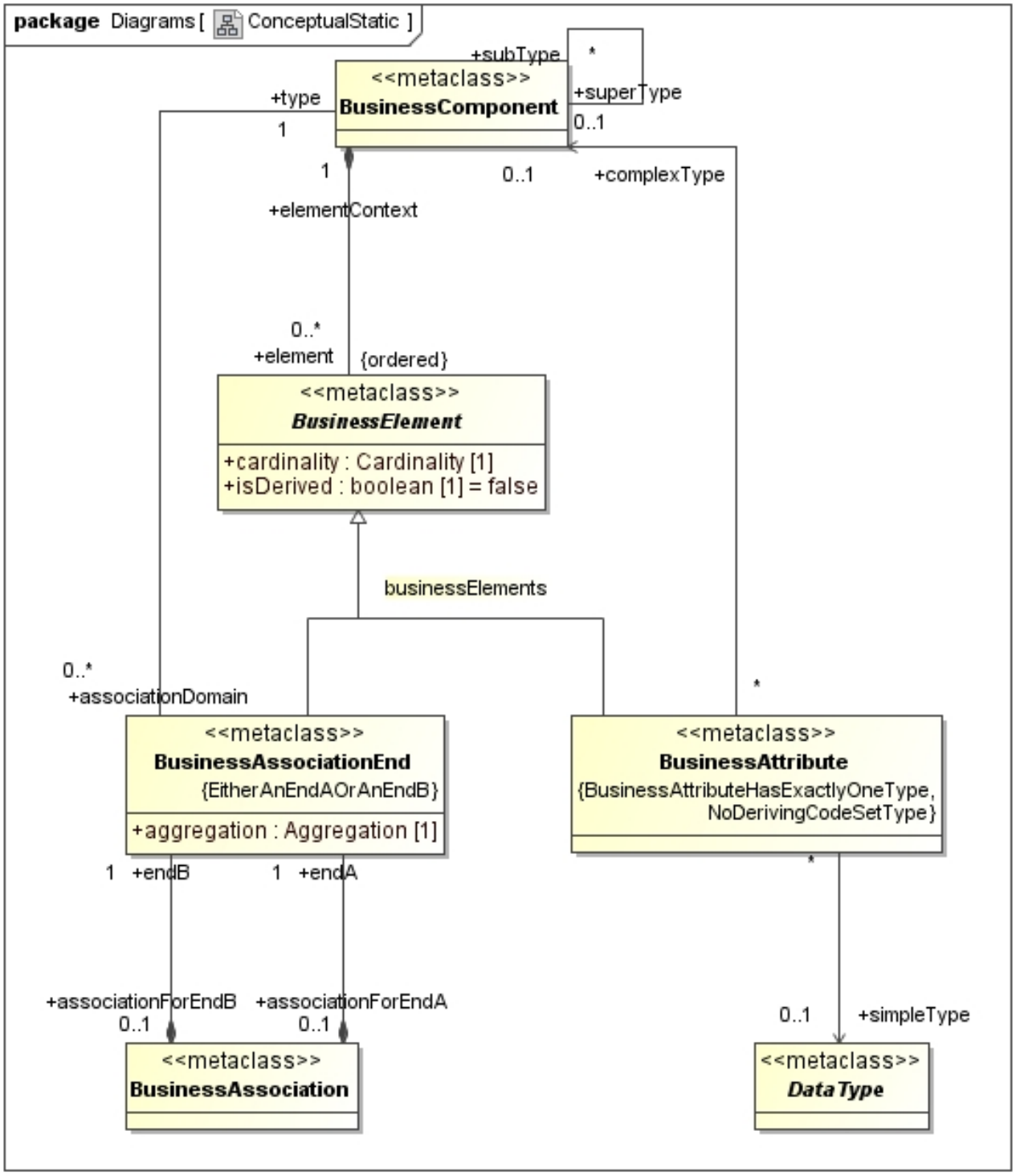


**Figure 5 — ConceptualStatic level**

#### 5.3.2.2 Identification of the requirements related to communication (dynamic part)

Define the dynamic part (i.e. the process description of the full interaction between all involved Participants) by specifying the details of the BusinessTransaction and MessageSet that will be developed. This means that the focus is moving from the “black box” perspective to a “white box” perspective. A “white box” perspective means that one looks into all the details of the BusinessTransaction and MessageSet. The focus is now on

defining the MessageFlows and MessageDefinitions that are needed to get the required information at the right time to the right business user.

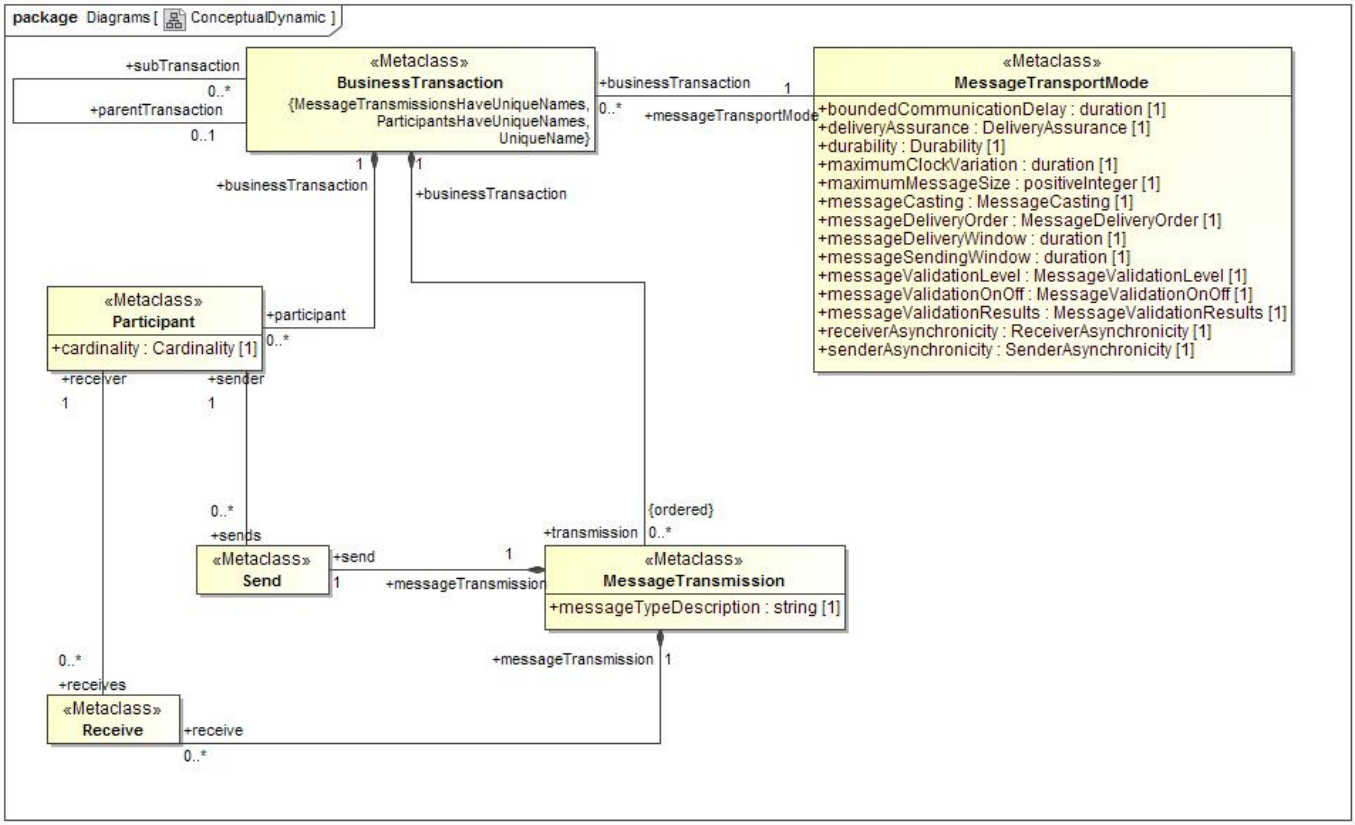


**Figure 6 — ConceptualDynamic level**

### 5.3.3 Logical level

The purpose of the Logical level is to create the MessageSets with their MessageDefinitions:

— the MessageSet is still characterized from a pure business perspective;

— the focus moves from defining the dynamics to defining the static part (i.e. the precise structure of the MessageInstances) of the BusinessTransaction and MessageSet, which is done in MessageDefinition Diagrams.

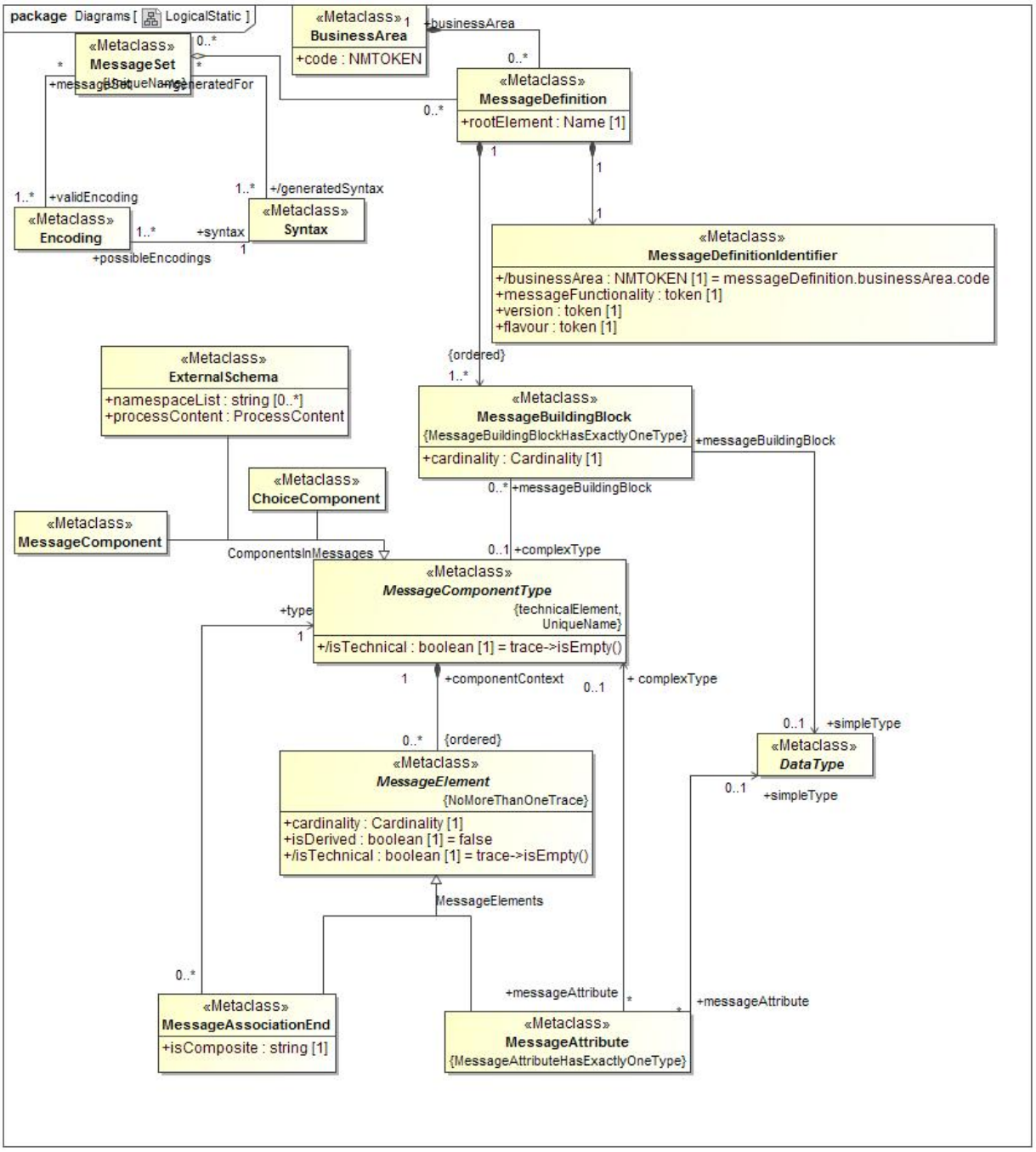


**Figure 7 — Logical level**

### 5.3.4 Physical level

The physical level delivers the physical implementation of MessageDefinitions and Constraints in an appropriate syntax, such as ISO 20022 XML. Specific design rules are used to build the physical representation of the MessageDefinitions from the deliverables of the Logical level.

The key deliverable of the Physical level is:

— a set of ISO 20022 Syntax Message Schemes;

— the Message Choreography.

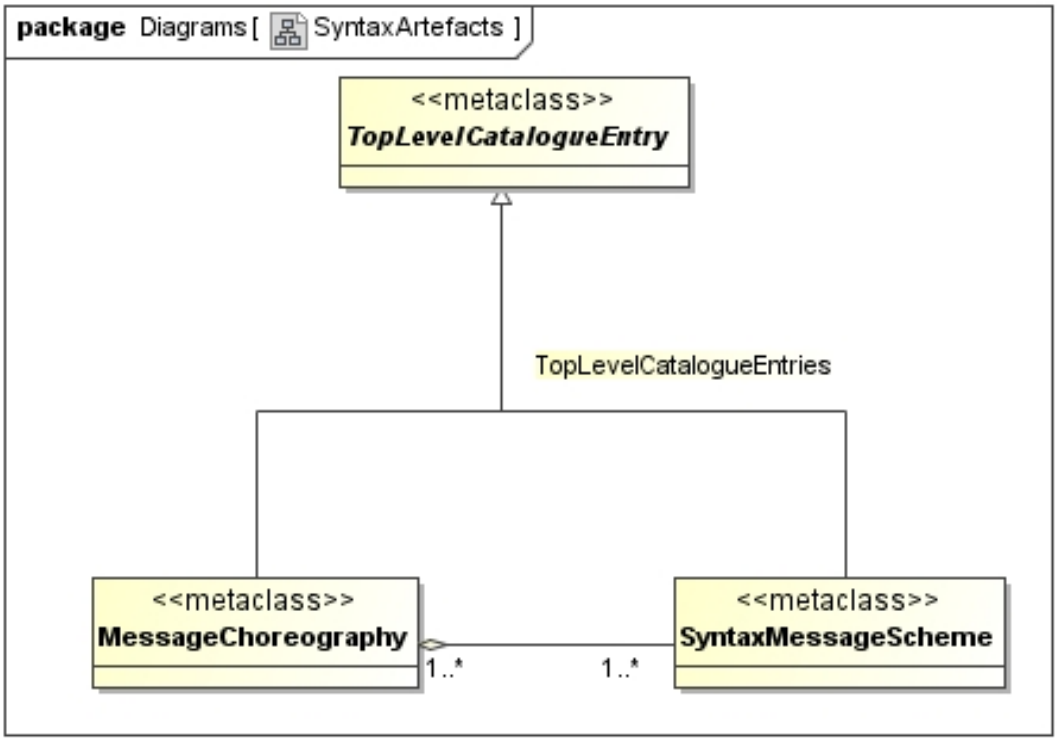


**Figure 8 — Syntax artefacts**

### 5.4 Models Transformation

Model transformation is inspired by the principles of OMG's MDA (Model Driven Architecture). To maintain consistency and facilitate realization, there is a transformation of the model at each level to its immediately subsequent level.

EXAMPLE The Conceptual level may be transformed to the Logical level.

The model at each level is transformed to the next level through being decorated with traces. Each trace is a map from one level to the next.

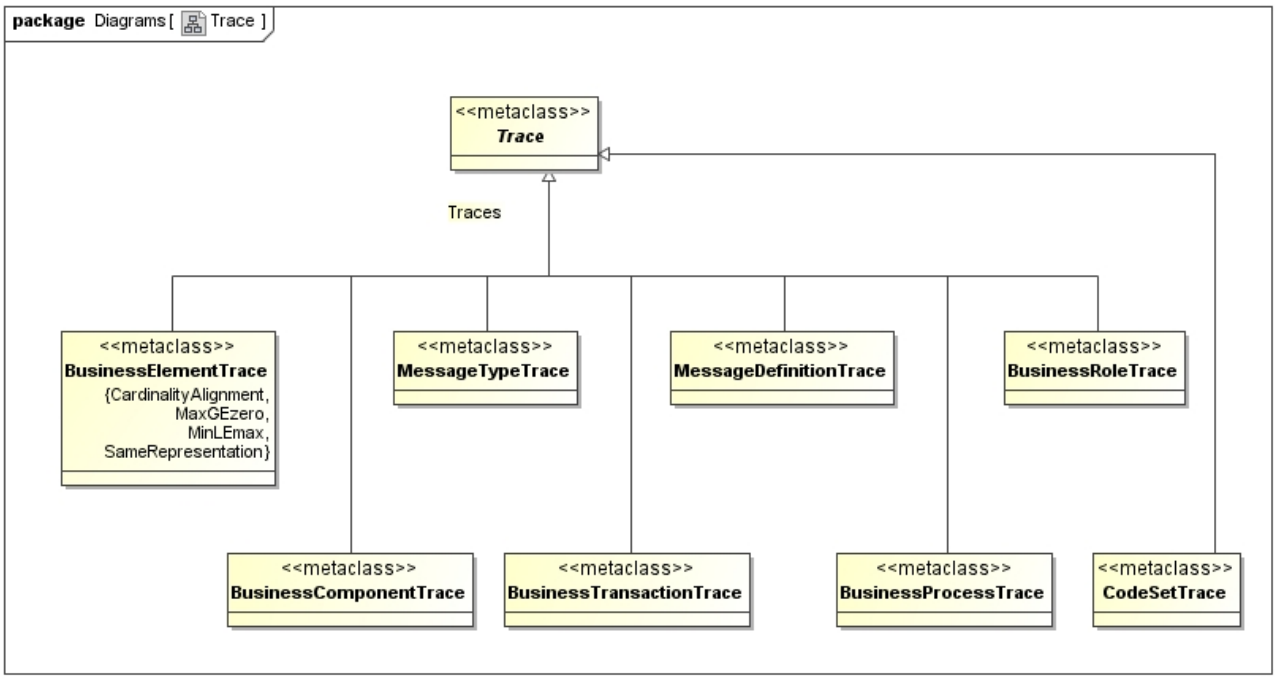


**Figure 9 — Traces**

**Table 2 — Transformation levels**

| Level name | Purpose |
|---|---|
| Scope to Conceptual | A map decorating a Scope level for the purpose of transformation to a Conceptual level. |
| Conceptual to Logical | A map decorating a Conceptual level for the purpose of transformation to a Logical level. |
| Logical to Physical | A map decorating a Logical level for the purpose of transformation to a Physical level. |

### 5.4.1 Scope to Conceptual

During the Conceptual level, the communication problem is described (the required information is scattered amongst the Participants). This means that for each BusinessProcess with its BusinessRoles, one or more BusinessTransactions with its Participants will have to be defined. A trace is defined for each Scope artefact to its corresponding Conceptual artefact.

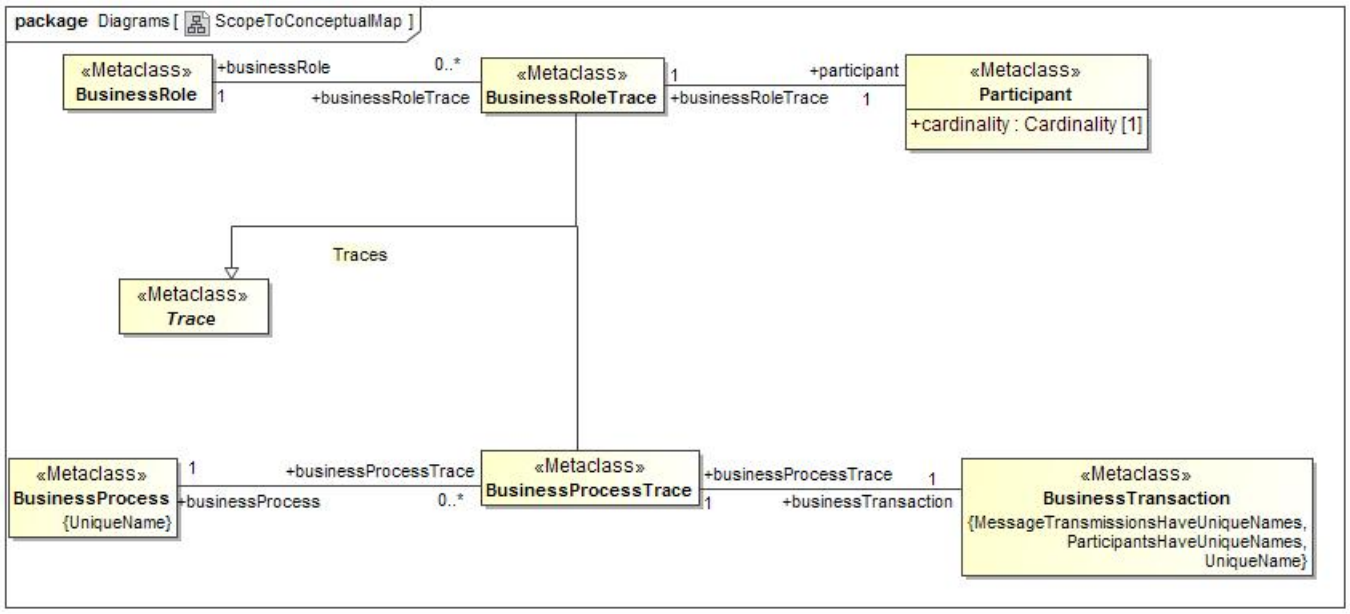


**Figure 10 — Scope to Conceptual level mapping**

### 5.4.2 Conceptual to Logical

MessageDefinitions are created by populating them with MessageComponentTypes. These MessageComponentTypes (which are contextual to the MessageDefinition in which they are used) may be derived from their more generic BusinessComponents. In such cases they shall be traced.

MessageTransmissions defined in BusinessTransaction Diagrams (identified as Signals in Sequence Diagrams) trace to MessageDefinitions.

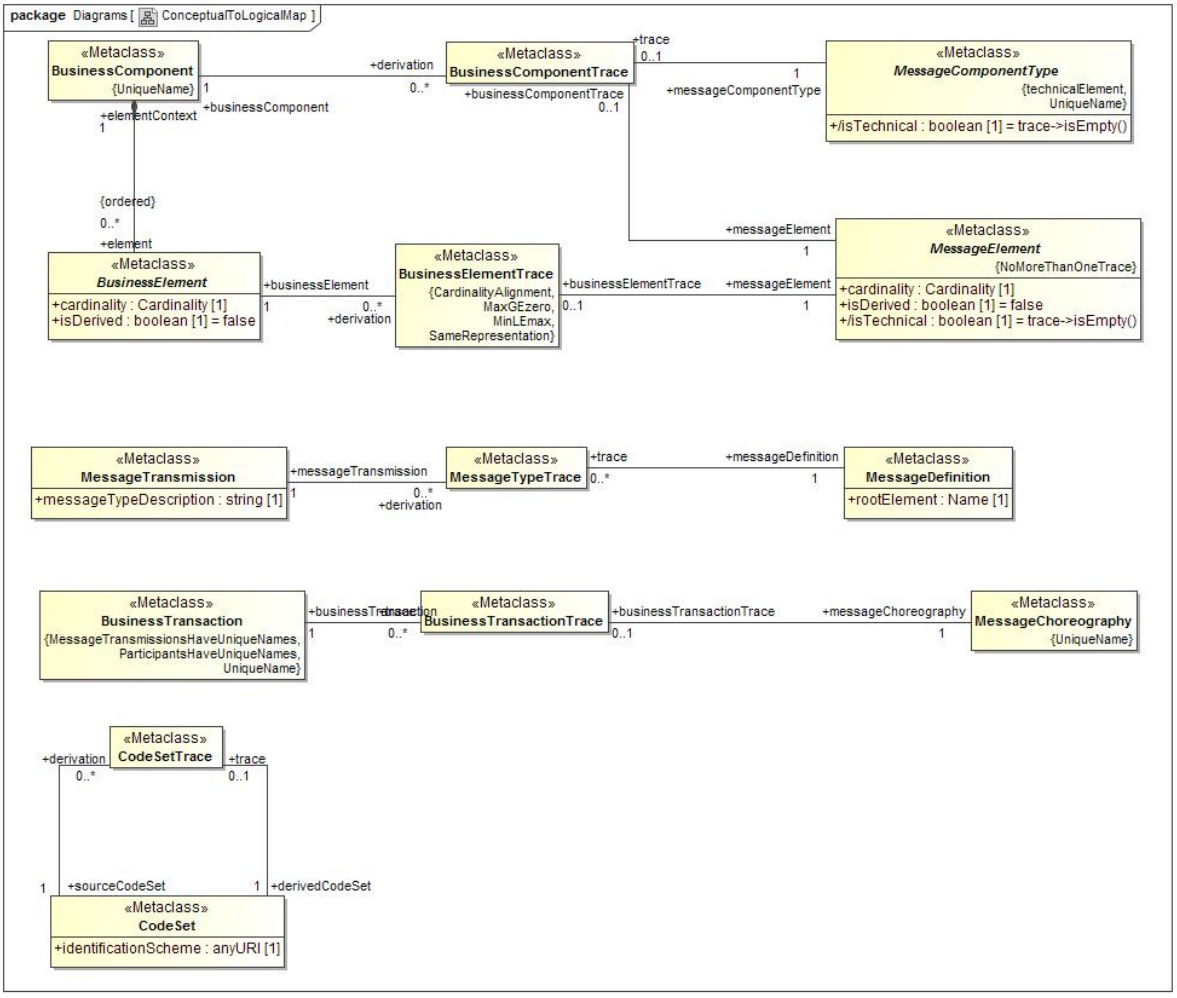


**Figure 11 — Conceptual to Logical level mapping**

### 5.4.3 Logical to Physical

This transformation is defined in ISO 20022-4.

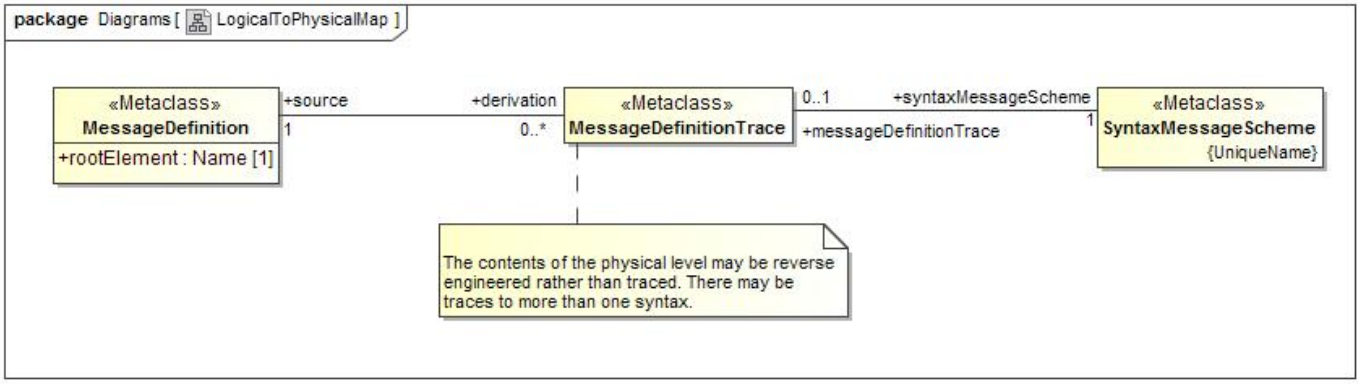


**Figure 12 — Logical to Physical level mapping**

# 6 Repository

## 6.1 ISO 20022 Repository structure

### 6.1.1 Overview

Figure 13 shows a high level view of the structure of the ISO 20022 Repository. As indicated, the ISO 20022 Repository consists of two major parts: the BusinessProcessCatalogue and the DataDictionary.

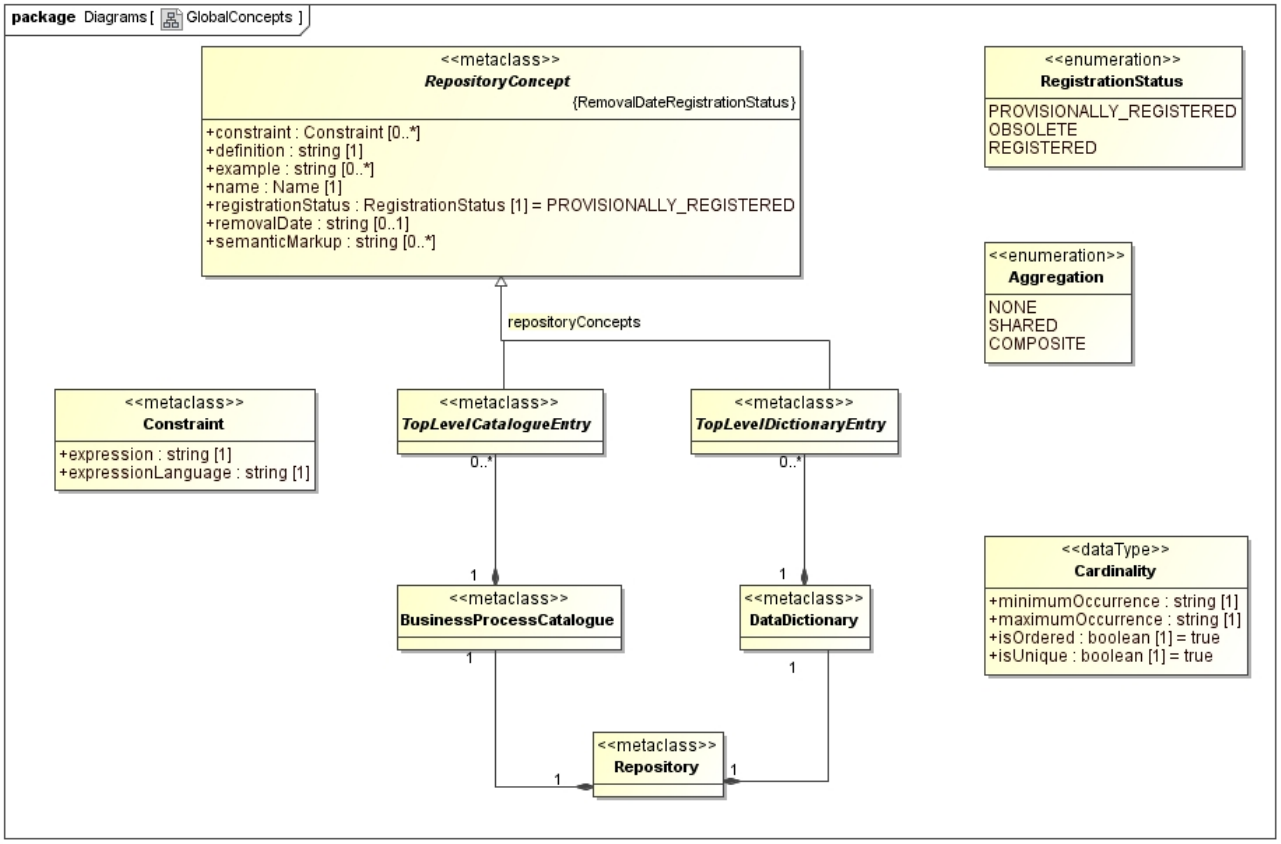


**Figure 13 — Global concepts**

The DataDictionary contains BusinessConcepts, MessageConcepts, Constraints and DataTypes. All these items are re-usable and are TopLevelDictionaryEntries. The DataDictionary as a whole is under release control.

The BusinessProcessCatalogue consists of TopLevelCatalogueEntries. The communication requirements and the interaction requirements in the various BusinessAreas are supported by BusinessTransactions, BusinessActivities and MessageDefinitions. All items that are stored in the BusinessProcessCatalogue are called Catalogue Items. Within the BusinessProcessCatalogue, the TopLevelCatalogueEntries are under release control. A BusinessProcessCatalogue release is always based on one single DataDictionary release.

The main relationships between Dictionary Items and Catalogue Items are as follows.

— Within the DataDictionary: MessageConcepts are derived from BusinessConcepts.

— Within the BusinessProcessCatalogue: BusinessTransactions support BusinessAreas.

— Between the BusinessProcessCatalogue and the DataDictionary: BusinessAreas are described using BusinessConcepts and BusinessTransactions are composed of MessageConcepts.

All Dictionary Items and Catalogue Items are uniquely identified, registered and managed within the Repository. The ISO 20022 Repository contains change history records, which contain the "change log" of those items, reflecting the life cycle of the items. Change history records are maintained and controlled by the Registration Authority.

The following change history record information is associated with every item:

— Change Type*:* identifies purpose of this change history record, in terms of item creation, amendment or deletion.

— Request By*:* identifies the institution or community of users that submitted the item's change request to the Registration Authority.

— Replaces*:* indicates, when applicable, the item that has been replaced by this item.

— Change description*:* provides the business justification to change the item.

— ChangeDate*:* date at which the change to the item has been registered.

## 6.2 DataDictionary

### 6.2.1 Overview

A DataDictionary Release contains Dictionary Items, namely BusinessConcepts, DataTypes and MessageConcepts. Each Dictionary Item is accompanied by some descriptive information and has a change history record containing information about its life cycle.

Figure 5 and Figure 7 show the main relationships between Dictionary Items:

— a BusinessComponent contains BusinessElements and Constraints;

— a MessageComponentType contains MessageElements and Constraints;

— a Code is a possible value of a CodeSet;

— a DataType defines the set of valid values of BusinessAttributes and of MessageElements;

— a DataType is based on a DataType representation;

— a MessageComponentType can be derived from a BusinessComponent;

— a MessageElement can be derived from a BusinessElement or BusinessComponent.

### 6.2.2 List of Dictionary Items

#### 6.2.2.1 BusinessConcepts

The DataDictionary contains the following types of BusinessConcept, which reflect the nature of the Dictionary Items that are defined and used during the Scope and Conceptual level activities. They form the basic items with which the Business Model of a BusinessArea is built.

— BusinessComponents;

— BusinessElements.

#### 6.2.2.2 DataTypes

The objective of a DataType is to specify unambiguously the value space of a BusinessAttribute, MessageBuildingBlock or a MessageAttribute.

XSD's Built-in DataTypes can be used directly as types of BusinessAttributes, MessageBuildingBlocks and MessageAttributes where, in contrast to the user-defined DataTypes, they cannot be further constrained. Every datatype imported from the XSD built-in library represents a set of values (i.e. its value space).

The DataType metaclass represents the set of all sets of values without identity. Therefore, datatypes imported from XSD built-in Datatypes library are instances of the DataType metaclass, which do not provide values for the metaproperties. Figure 14 illustrates the allowed instances and the relationship between the datatypes and their metaclass.

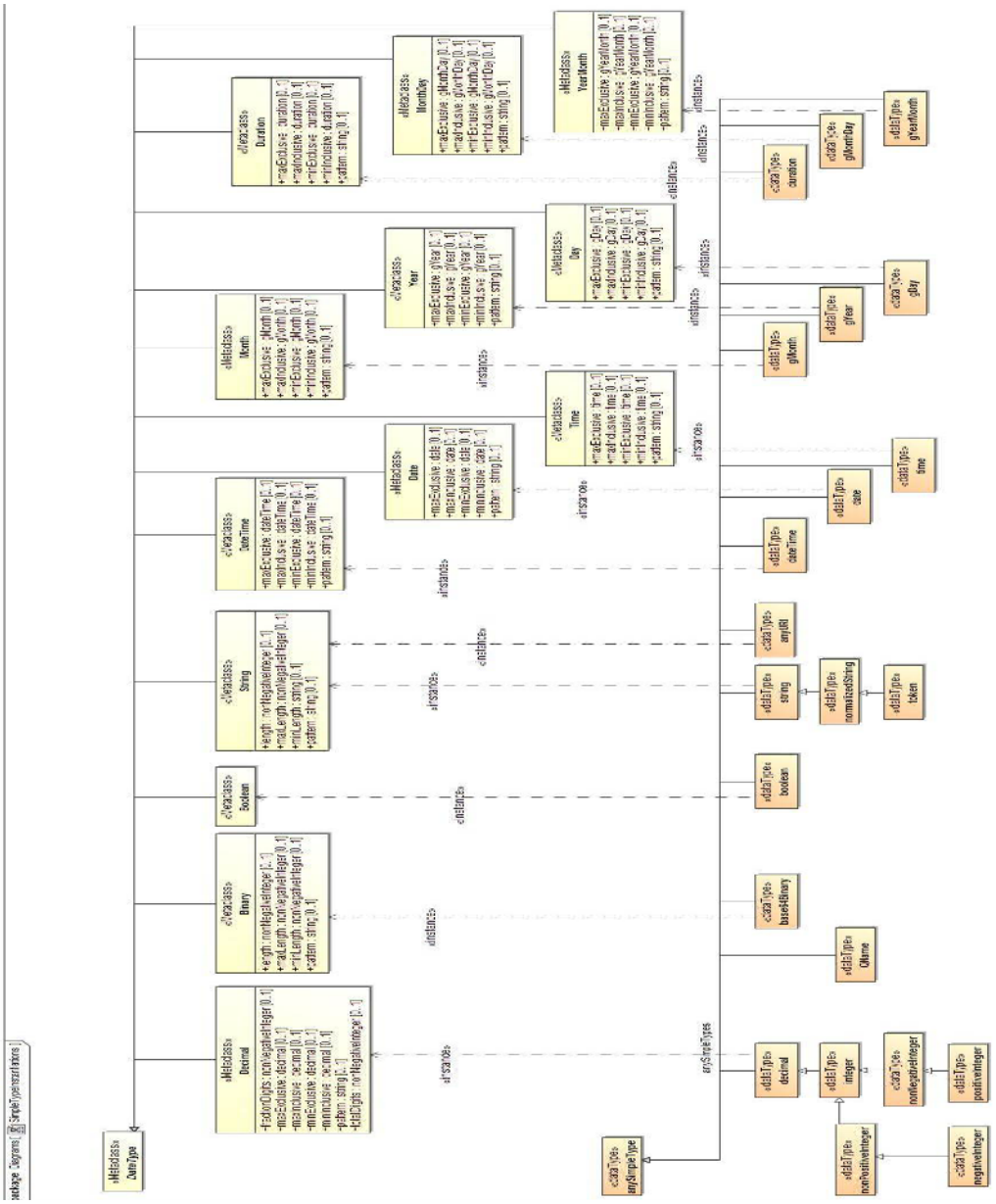

**Figure 14 — XSD built-in DataType instantiations**

The user-defined DataTypes are categorized in a limited number of datatype representations, such as Amount, IdentifierSet, Quantity, CodeSet, Date, Time, Text, etc. The full list of DataType representations is defined in the metamodel.

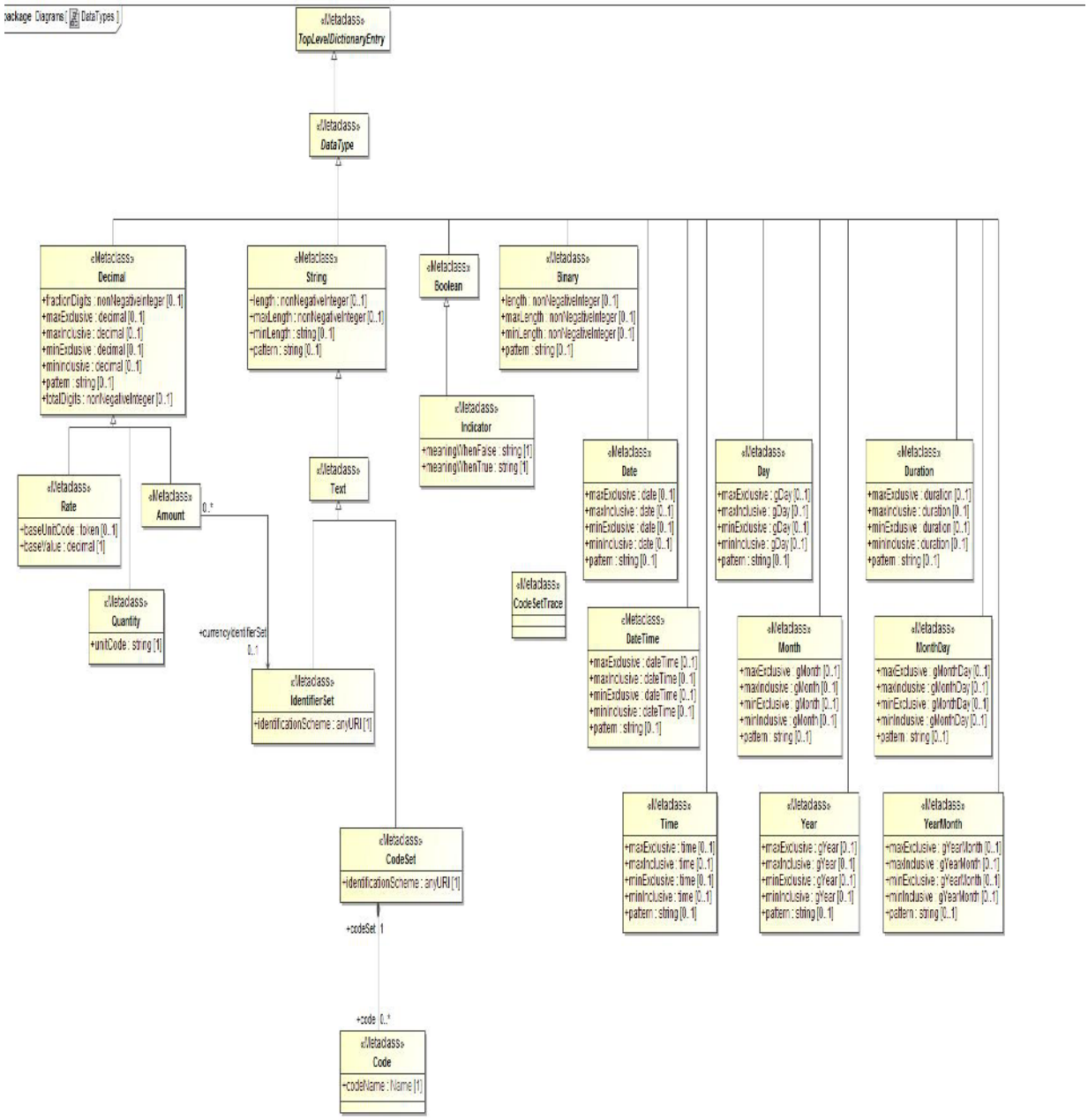


**Figure 15 — User-defined DataTypes**

Each DataType representation defines the following information.

a) The XSD Built-in DataType that will be used for all DataTypes based on this DataType representation, and therefore its value space, e.g.:

— DataTypes based on DataType representation “Text” will restrict XSD String;

— DataTypes based on DataType representation “Amount” will restrict XSD Decimal.

b) The additional information that shall be specified to distinguish DataTypes based on the same DataType representation. This additional information will constrain the value space for a particular DataType. This can be done in the following two ways.

— Defining explicitly the set of possible values, by exhaustive enumeration or by referencing a list. For the DataType representation “CodeSet”, the Registration Authority will either use an existing set of Codes or the Registration Authority will define a new set of Codes within the ISO 20022 Repository. In the latter case, all Codes will be of 1 up to 4 alphanumeric characters where the first shall be upper alphabetic.

— Specifying a format restriction, e.g. by defining the allowed length of a string, using the DataType properties.

EXAMPLE 1 DataTypes based on DataType representation “IdentifierSet” can specify the list that contains the possible values of this identifier, e.g. the DataType “ISIN” contains its possible values in the “ISIN directory”.

EXAMPLE 2 DataTypes based on DataType representation “Text” can specify the maximum length that is allowed, e.g. the DataType “Max35Text”, which is used, among others, for the street name, has a maximum of 35 characters.

The list of possible values for DataTypes that have the DataType representation “CodeSet” will be included in the DataDictionary if the set of Codes is maintained by the ISO 20022 Registration Authority.

#### 6.2.2.3 MessageConcepts

The DataDictionary contains the following types of Message Concept, which reflect the nature of the Dictionary Items that are defined and used during the Logical level activities. They form the basic items of MessageDefinitions.

— MessageComponentTypes are the re-usable Dictionary Items with which the MessageDefinitions shall be built. A MessageComponentType, when it has a business meaning, is derived from one single BusinessComponent. It can be considered as a “view” on that BusinessComponent that will be used in MessageDefinitions.

  Several MessageComponentTypes can be derived from the same BusinessComponent. These MessageComponentTypes will be different because of their specific subset of MessageElements or because of specific constraints such as cross-element Constraints or multiplicity constraints.

  In some cases a MessageComponentType can be based on a set of related BusinessComponents or can even be defined for message-specific reasons without being derived from any BusinessComponent at all.

— MessageElements: these are usually derived from the BusinessElements of the BusinessComponent corresponding to the MessageComponentType. There might be situations where MessageElements in one MessageComponentType come from multiple related BusinessComponents. The MessageElements will then be linked to the relevant BusinessElement or BusinessComponent.

  When the MessageElement has no business meaning it is not derived from any BusinessElement at all.

  If a MessageElement is derived from other MessageElements, then a Constraint should specify how this MessageElement is derived.

  The value space of a MessageElement is either defined by a DataType or by another MessageComponentType.

### 6.2.3 Dictionary Item Registration Status

Each Dictionary Item is assigned a “Registration Status”. The Registration Status can take the following values:

— PROVISIONALLY REGISTERED: the Dictionary Item is pending final approval (see ISO 20022-7). This enables the Registration Authority to inform a community of users of Dictionary Items that the item will potentially become "REGISTERED" in the near future.

— REGISTERED: the ISO 20022 compliant Dictionary Item is approved and can be used.

— OBSOLETE [<REMOVAL-DATE>]: the Dictionary Item is no longer considered as ISO 20022 compliant and may no longer be used for registering updates to the ISO 20022 Repository. The Dictionary Item will, however, be kept in the DataDictionary as long as it is used in other Dictionary or Catalogue Items. The optional <REMOVAL-DATE> enables the Registration Authority to inform a community of users that the Dictionary Item will be physically removed from the DataDictionary at the date specified as <REMOVAL-DATE>.

### 6.2.4 Dictionary Items description information

The semantics of each Dictionary Item is defined by the following information[1)];

— Name: official name of the Dictionary Item (uniqueness is defined in the metamodel). It may include versioning information.

— Definition*:* precise description of the meaning of the Dictionary Item.

— Removal date: the date at which a Dictionary Item having an "OBSOLETE" Registration status is removed from the Repository.

— SemanticMarkup: markup of elements of the Dictionary with semantic metadata.

— Examples: examples of the use of the Dictionary Item in its business context.

— Multiplicity: indicates whether the Dictionary Item is mandatory, optional and/or repetitive. This information is only applicable to BusinessElements in BusinessComponents and to MessageElements in MessageComponentTypes.

— Constraint: a semantic condition or restriction.

### 6.2.5 DataDictionary life cycle

On a regular basis, the Registration Authority makes publicly available the current release of the DataDictionary. This published DataDictionary contains all Dictionary Items, i.e. with status PROVISIONALLY REGISTERED, REGISTERED or OBSOLETE [<REMOVAL-DATE>][2)] . This allows the Registration Authority to inform communities of users of Dictionary Items that will either become registered, or will be removed from the Dictionary in the near future.

At any time, only the latest release of the DataDictionary is available as the official point of reference. It replaces any previously released DataDictionary and contains at least all Dictionary Items that are referenced in the current issue of the BusinessProcessCatalogue.

The Registration Authority will also publish an archive of Message Definitions with status OBSOLETE and a <REMOVAL-DATE> less than than the current release publication date. This allows a user community to find information about previous versions of Message Definitions that are no longer officially supported.

---

1) This list of information is not exhaustive but is provided in the scope of the information that might be required for the submission requests to the Registration Authority.

2) "<REMOVAL-DATE>" is greater than the release publication date.

Change history records are published at the same time as the DataDictionary. The change history records reflect the evolution of the Dictionary Items.

## 6.3 BusinessProcessCatalogue

### 6.3.1 Overview

The BusinessProcessCatalogue contains BusinessAreas. A BusinessArea can be described in more details using Business Processes. The communication requirements and the interaction requirements in the various BusinessAreas and Business Processes are supported by BusinessTransactions, which include a detailed description of the possible message flows in BusinessTransaction Diagrams. A BusinessTransaction Diagram contains one or more MessageTransmissions. Each MessageTransmission is described in a MessageDefinition, which is also converted into an ISO 20022 Syntax Message Representation.

Every MessageDefinition is thus associated with the following context information when it is used to support the communication requirements of a business activity:

— a specific BusinessTransaction Diagram, itself being used in support of a BusinessTransaction;

— a specific BusinessArea, detailed in a specific BusinessProcess.

All this information is required to understand the precise function, role and application of a MessageDefinition in the considered BusinessProcess and/or BusinessArea.

### 6.3.2 List of BusinessProcessCatalogue Items

The following items may be contained in the BusinessProcessCatalogue:

— BusinessArea with its MessageDefinitions (Diagram);

— BusinessProcess (Diagram) with its BusinessRoles;

— BusinessTransaction (Diagram) with its Participants;

— MessageSet with its MessageDefinitions (Diagram);

— SyntaxMessageScheme;

— MessageChoreography;

— MessageTransportMode.

### 6.3.3 Catalogue Item Registration Status

Each Catalogue Item is assigned a “Registration Status”. The Registration Status may take the following values:

— `PROVISIONALLY REGISTERED`: the Catalogue Item is pending final approval (see ISO 20022-7). Note that, unlike for DataDictionary Items, Catalogue Items with this status are not made publicly available;

— `REGISTERED`: the ISO 20022 compliant Catalogue Item is approved;

— `OBSOLETE [<REMOVAL-DATE>]`: the Catalogue Item is no longer considered as ISO 20022 compliant and may no longer be used for registering updates to the ISO 20022 Repository. The Catalogue Item will, however, be kept in the BusinessProcessCatalogue as long as it is used. The optional `<REMOVAL-DATE>` enables the Registration Authority to inform a community of users that the Catalogue Item will be physically removed from the BusinessProcessCatalogue at the date specified as `<REMOVAL-DATE>`.

#### 6.3.4 Catalogue Item description information

The semantics of each Catalogue Item is defined by the following information[3)]:

- Name*:* official name of the Catalogue Item, which may include versioning information.
- Definition: precise description of the meaning of the Catalogue Item.
- Removal date: the date at which a Catalogue Item having an "OBSOLETE" Registration status is removed from the Repository.
- Example*: e*xamples of the use of the Catalogue Item in its business context.
- semanticMarkup: markup of elements of the Catalogue with semantic metadata.
- constraint: a semantic condition or restriction.

#### 6.3.5 BusinessProcessCatalogue life cycle

On a regular basis, the Registration Authority makes publicly available the current release of the BusinessProcessCatalogue. This published BusinessProcessCatalogue contains all Catalogue Items with status REGISTERED or OBSOLETE [<REMOVAL-DATE>]. Catalogue Items that have the status <PROVISIONALLY REGISTERED> will not be accessible via the public BusinessProcessCatalogue. The experts nominated by ISO/TC 68 (see ISO 20022-7) who are responsible for the approval of these Catalogue Items will, however, get the necessary documentation in an alternative, off-line way via the Registration Authority.

At any time, only the latest release of the BusinessProcessCatalogue is available as the official point of reference. It replaces any previously released BusinessProcessCatalogue and is always associated with the latest DataDictionary Release. Note that a BusinessTransaction may only have the REGISTERED status if all its related Catalogue Items and Dictionary Items have the status REGISTERED or OBSOLETE with a <REMOVAL-DATE> greater than the release publication date.

The Registration Authority will also publish archive Releases of the BusinessProcessCatalogue. This allows a user community to find information about Catalogue Items that are no longer officially supported.

Change history records are published at the same time as the BusinessProcessCatalogue. The change history records reflect the evolution of the Catalogue Items.

## 7 Registration

### 7.1 General

This clause provides specifications about the submission requests to the Registration Authority for updates to the DataDictionary or BusinessProcessCatalogue.

The following information is provided:

- the required format of the input;
- the means by which the input can be conveyed to the Registration Authority.

Further information on submission of requests and related Registration Authority service levels can be found in ISO 20022-7.

3) This list is not exhaustive but is provided in the scope of the information that might be required for the submission requests to the Registration Authority.

### 7.2 Submission format

The submission will include new or updated Business Processes, Business Transactions and Message Definitions. The models shall be compliant with the metamodel and, if submitted in UML, with ISO 20022-2 and ISO 20022-3. New or updated Repository items shall be identified.

The format of the submission is to be agreed with the Registration Authority. Acceptable formats and related conventions will be published on the ISO 20022 website.

### 7.3 Submission media

An e-mail, sent to the Registration Authority e-mail address, will be the authorized submission media.

With the evolution of technology, the Registration Authority might accept other submission media. The full list of valid submission means shall be maintained on the ISO 20022 website.

## 8 Repository Access

### 8.1 General

It is the responsibility of the Registration Authority to provide public access to the ISO 20022 Repository information. This access will give the users the possibility to search, identify and get the full description of the available ISO 20022 DataDictionary, BusinessTransactions and MessageSets.

### 8.2 Repository output types

The Registration Authority shall provide access to the ISO 20022 Repository information via the following output types.

a) Interactive search/queries on:

- the DataDictionary, giving access to all DataDictionary Items and exploiting the relations between DataDictionary Items;
- the BusinessProcessCatalogue, giving access to all Catalogue Items and exploiting the relations between Catalogue Items.

b) A set of predefined downloadable extracts providing information about:

- the DataDictionary, listing all DataDictionary Items with all relevant information, e.g. Name, Definition, etc.;
- the BusinessProcessCatalogue (Business Processes and their related BusinessTransactions, BusinessTransaction Diagrams, MessageDefinitions and ISO 20022 Syntax Message Representations).

### 8.3 Output format

The interactive queries shall be provided free of charge via a “web query” interface on the ISO 20022 website.

The downloadable extracts shall be provided free of charge in at least one of the following formats via the ISO 20022 website:

- RTF;
- HTML;

— PDF.

The Registration Authority shall also provide repository extracts in XMI and may agree, on a case-by-case basis, to deliver ISO 20022 Repository information in other structured languages or processable formats. The Registration Authority may charge a reasonable fee to the requesting community of users for these deliverables.

# Annex A
(normative)

# Type library

## A.1 General

This International Standard's Type Library is used in both its Metamodel and in its Models.

The type library uses MOF 1.4 and is expressed in XMI 2.1.

## A.2 Type Library Details

### A.2.1 Package ISO20022::TypeLibrary

| | |
|---|---|
| Applied Stereotype | |

The Standards Type Library. This is used in both the Metamodel and the Models.

### A.2.2 Package ISO20022::TypeLibrary::Enumerations

#### A.2.2.1 Enumeration Aggregation

**Description**

Enumeration of the different kinds of "has a" relationship supported by the metamodel.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::Aggregation |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| NONE | The relationship between two RepositoryConcepts is expressed through a simple association. |
| SHARED | A kind of relationship between two RepositoryConcepts whereby one component may belong to multiple aggregates.<br><br>EXAMPLE A person may belong to different teams. If a team disappears, the person still exists. |
| COMPOSITE | A kind of relationship between two RepositoryConcepts expressing that one of the concepts cannot exist without the other.<br><br>EXAMPLE An Account has an AccountOwner. If a person who has an account ceases to exist, so will its account cease to exist. |

### A.2.2.2 Enumeration DeliveryAssurance

**Description**

Characteristic from the MessageTransport, specifying the degree to which the sending MessagingEndpoint is assured that a TransportMessage will be delivered.

NOTE The MessageTransportSystem is responsible for implementing this characteristic and might therefore decide to achieve this by republishing Transport Messages. The Idempotent Behaviours apply.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::DeliveryAssurance |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| AT_LEAST_ONCE | The receiving Messaging Endpoint receives the TransportMessage at least once.<br><br>NOTE The delivery of the TransportMessage is highly assured. If the TransportMessage cannot be delivered, then the sending Messaging Endpoint shall receive notification of an error. |
| EXACTLY_ONCE | The receiving Messaging Endpoint receives the TransportMessage exactly once.<br><br>NOTE If the TransportMessage cannot be delivered, then the sending Messaging Endpoint shall receive notification of an error. |
| AT_MOST_ONCE | The receiving MessagingEndpoint receives the TransportMessage at most once. There is no assurance a TransportMessage will be delivered.<br><br>NOTE The MessageTransportSystem does not deliver notification errors for non-delivery. If the TransportMessage cannot be delivered, then the sending MessagingEndpoint shall not receive notification of an error. |

### A.2.2.3 Enumeration durability

**Description**

Characteristic from the MessageTransport, indicating whether the MessageTransportSystem safely retains a TransportMessage until it has been received by the destination MessagingEndpoint.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::Durability |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| DURABLE | The TransportMessage is kept available indefinitely until the message is delivered to the destination MessagingEndpoint. The TransportMessage is only kept available if it is ready for delivery within the BoundedCommunicationDelay; otherwise it expires like every other non-durable message. |
| PERSISTENT | The TransportMessage is kept available until it is delivered to the receiving MessagingEndpoint or until it is expired because the BoundedCommunicationDelay is exceeded. |

| | |
|---|---|
| TRANSIENT | The TransportMessage is not kept available and will not be delivered if the receiving MessagingEndpoint is not available. |

#### A.2.2.4 Enumeration MessageCasting

**Description**

Characteristic from the MessageTransport, indicating how receiving MessagingEndpoints may be addressed in a TransportMessage.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::MessageCasting |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| UNICAST | TransportMessages are addressed to a single receiving MessagingEndpoint. |
| MULTICAST | TransportMessages are addressed to nought to many receiving MessagingEndpoints. |
| BROADCAST | TransportMessages are addressed to a single Broadcast List. |
| ANYCAST | TransportMessages may be Multicast or Broadcast.<br><br>NOTE 1 The first MessagingEndpoint to consume the TransportMessage causes it to be removed from all other MessagingEndpoints.<br><br>NOTE 2 At most only one MessagingEndpoint receives the message. |

#### A.2.2.5 Enumeration MessageDeliveryOrder

**Description**

Characteristic from the MessageTransport, indicating to what extent TransportMessages from a sending MessagingEndpoint arrive in the order in which they were sent at the receiving MessagingEndpoints.

NOTE 1 The sending order is defined as the order in which the messages arrive at the MessageTransportSystem.

NOTE 2 The time at which a TransportMessage is sent from a MessagingEndpoint to the MessagingTransportSystem is the same time as the TransportMessage arrives at the MessageTransportSystem.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::MessageDeliveryOrder |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| EXPECTED_CAUSAL_ORDER | The receipt order of TransportMessages at all receivers is preserved across all sending MessagingEndpoints. |

| | |
|---|---|
| FIFO_ORDERED | At every receiver, the order of receipt of TransportMessages is preserved for each sending MessagingEndpoint but not across sending MessagingEndpoints. |
| UNORDERED | TransportMessages may arrive in any order at the receiving MessagingEndpoints. The only limitation is the Bounded Communication Delay. |

#### A.2.2.6 Enumeration MessageValidationLevel

**Description**

Characteristic from the MessageTransport, indicating the level of validation that is required by the MessageTransportSystem.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::MessageValidationLevel |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| NO_VALIDATION | The MessageInstance is not validated. |
| SYNTAX_VALID | The MessageInstance has its syntax validated,<br><br>e.g. for XML messages, this would mean testing the XML is well-formed. |
| SCHEMA_VALID | The MessageInstance is Syntax Valid plus validated against the Syntax Message Scheme,<br><br>e.g. for XML messages, this would mean testing the XML is well-formed and valid against the ISO 20022 XML Schema. |
| MESSAGE_VALID | The MessageInstance is Schema Valid plus validated against the MessageRules. |
| RULE_VALID | The MessageInstance is Message Valid plus validated against the BusinessRules. |
| MARKET_PRACTICE_VALID | The MessageInstance is Message Valid plus validated against the Market Practices. |
| BUSINESS_PROCESS_VALID | The MessageInstance is Message Valid plus validated against the MessageChoreography. |
| COMPLETELY_VALID | The MessageInstance is Message Valid plus validated against all Rules and MarketPractices. |

#### A.2.2.7 Enumeration MessageValidationOnOff

**Description**

Characteristic from the MessageTransport, specifying whether the MessageTransportSystem validates the MessageInstance with respect to SyntaxMessageScheme, Rules, the MarketPractices, and the MessageChoreography.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::MessageValidationOnOff |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| VALIDATION_ON | All MessageInstances are validated by the MessageTransportSystem. |
| VALIDATION_OFF | MessageInstances are not validated by the MessageTransportSystem. |

#### A.2.2.8 Enumeration MessageValidationResults

**Description**

Characteristic from the MessageTransport, specifying the behaviour of the MessageTransportSystem as a result of MessageValidation.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::MessageValidationResults |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| REJECT | Invalid messages cause a rejection TransportMessage to be sent to the sending MessagingEndpoint, and the invalid MessageInstance is not delivered to any other MessagingEndpoint. Valid messages are delivered to their destinations and marked as valid in the TransportMessage. |
| REJECT_AND_DELIVER | Invalid messages shall cause a rejection TransportMessage to be sent to the sending MessagingEndpoint, and the invalid MessageInstance is marked as invalid in the TransportMessage and delivered to its destination MessagingEndpoints. Valid messages are delivered to their destinations and marked as valid in the TransportMessage. |
| DELIVER | Invalid messages do not cause a rejection TransportMessage to be sent to the sending MessagingEndpoint. The invalid MessageInstance is delivered to its destination MessagingEndpoints. Valid messages are delivered to their destinations and marked as valid in the TransportMessage. |

#### A.2.2.9 Enumeration ProcessContent

**Description**

Specifies how a contentmodel shall be validated.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::ProcessContent |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| SKIP | The contentmodel shall not be validated. |
| STRICT | The contentmodel shall be validated against its specification. |
| LAX | The contentmodel shall be validated on a can-do basis. |

#### A.2.2.10 Enumeration ReceiverAsynchronicity

**Description**

Characteristic from the MessageTransport, indicating whether a receiving MessagingEndpoint blocks the sending and receipt of other TransportMessages until it sends a response to this TransportMessage.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::ReceiverAsynchronicity |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| ENDPOINT_SYNCHRONOUS | The receiving MessagingEndpoint blocks the receipt and processing of other TransportMessages while processing the current TransportMessage. |
| CONVERSATION_SYNCHRONOUS | The receiving MessagingEndpoint blocks the sending and receipt of other TransportMessages within the conversation, in which the TransportMessage was sent, while waiting for a response to this sent TransportMessage. |
| ASYNCHRONOUS | The receiving MessagingEndpoint shall not block the receipt or processing of other TransportMessages while processing the current TransportMessage. |

#### A.2.2.11 Enumeration RegistrationStatus

**Description**

The enumeration of the standing of a RepositoryConcept in the Repository.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::RegistrationStatus |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| PROVISIONALLY_REGISTERED | Status of a RepositoryConcept, showing it is pending final approval. |
| OBSOLETE | Status of a RepositoryConcept, showing it is still compliant but cannot be used in new ISO 20022 compliant developments.<br><br>NOTE Combined with the removalDate, it means the date as of which this RepositoryConcept is no longer considered as compliant. It may no longer be used for registering updates to the Repository and will be removed from the Repository at the date specified in removalDate. |
| REGISTERED | Status of a RepositoryConcept, showing it is ISO 20022 compliant, approved by the Registration Authority and can be used. |

#### A.2.2.12 Enumeration SenderAsynchronicity

**Description**

Characteristic from the MessageTransport, indicating whether a sending Messaging Endpoint blocks after sending a TransportMessage to the MessageTransportSystem while waiting for a response from a MessagingEndpoint.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::Enumerations::SenderAsynchronicity |
| Visibility | public |

**Owned literals**

| Literal name | Description |
|---|---|
| ENDPOINT_SYNCHRONOUS | The sending MessagingEndpoint blocks the sending and receipt of other TransportMessages while waiting for a response to the sent TransportMessage. |
| CONVERSATION_SYNCHRONOUS | The sending MessagingEndpoint blocks the sending and receipt of other TransportMessages within the conversation, in which the TransportMessage was sent, while waiting for a response to this sent TransportMessage. |
| ASYNCHRONOUS | The sending MessagingEndpoint shall not block the sending or receipt of other TransportMessages while waiting for a response to the sent TransportMessage. |

### A.2.3 Package ISO20022::TypeLibrary::XMLSchema

| | |
|---|---|
| Applied Stereotype | |

**Description**

#### A.2.3.1 Datatype anySimpleType

**Description**

XSD data type defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anyType |

#### A.2.3.2 Datatype anyType

**Description**

XSD data type 'anyType' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::anyType |
| Visibility | public |
| Supertype | |

#### A.2.3.3 Datatype anyURI

**Description**

XSD data type 'anyURI' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::anyURI |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::String |

#### A.2.3.4 Datatype base64Binary

**Description**

XSD data type 'base64Binary' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::base64Binary |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Binary |

#### A.2.3.5 Datatype boolean

**Description**

XSD data type 'boolean' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::boolean |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Boolean |

#### A.2.3.6 Datatype byte

**Description**

XSD data type 'byte' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::byte |

| | |
|---|---|
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::short |

#### A.2.3.7 Datatype date

**Description**

XSD data type 'date' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::date |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Date |

#### A.2.3.8 Datatype dateTime

**Description**

XSD data type 'dateTime' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::dateTime |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::DateTime |

#### A.2.3.9 Datatype decimal

**Description**

XSD data type 'Decimal' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::decimal |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Decimal |

#### A.2.3.10 Datatype double

**Description**

XSD data type 'double' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::double |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |

#### A.2.3.11 Datatype duration

**Description**

XSD data type 'duration' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::duration |
| Visibility | public |

| | |
|---|---|
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Duration |

#### A.2.3.12 Datatype ENTITIES

**Description**

XSD data type 'ENTITIES' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::ENTITIES |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::ENTITY |

#### A.2.3.13 Datatype ENTITY

**Description**

XSD data type 'ENTITY' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::ENTITY |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::NCName |

#### A.2.3.14 Datatype float

**Description**

XSD data type 'float' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::float |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |

#### A.2.3.15 Datatype gDay

**Description**

XSD data type 'gDay' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::gDay |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Day |

#### A.2.3.16 Datatype gMonth

**Description**

XSD data type 'gMonth' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::gMonth |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Month |

#### A.2.3.17 Datatype gMonthDay

**Description**

XSD data type 'gMonthDay' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::gMonthDay |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::MonthDay |

#### A.2.3.18 Datatype gYear

**Description**

XSD data type 'gYear' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::gYear |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Year |

#### A.2.3.19 Datatype gYearMonth

**Description**

XSD data type 'gYearMonth' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::gYearMonth |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::YearMonth |

#### A.2.3.20 Datatype hexBinary

**Description**

XSD data type 'hexBinary' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::hexBinary |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |

#### A.2.3.21 Datatype ID

**Description**

XSD data type 'ID' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::ID |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::NCName |

#### A.2.3.22 Datatype IDREF

**Description**

XSD data type 'IDREF' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::IDREF |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::NCName |

#### A.2.3.23 Datatype IDREFS

**Description**

XSD data type 'IDREFS' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::IDREFS |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::IDREF |

#### A.2.3.24 Datatype int

**Description**

XSD data type 'int' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::int |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::long |

#### A.2.3.25 Datatype integer

**Description**

XSD data type 'integer' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::integer |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::decimal |

#### A.2.3.26 Datatype language

**Description**

XSD data type 'language' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::language |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::token |

#### A.2.3.27 Datatype long

**Description**

XSD data type 'long' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::long |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::integer |

#### A.2.3.28 Datatype Name

**Description**

XSD data type 'Name' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::Name |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::token |

#### A.2.3.29 Datatype NCName

**Description**

XML "non-colonised" Names (i.e. names that do not contain a colon ":").

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::NCName |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::Name |

#### A.2.3.30 Datatype negativeInteger

**Description**

XSD data type 'negativeInteger' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::negativeInteger |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::nonPositiveInteger |

#### A.2.3.31 Datatype NMTOKEN

**Description**

XSD data type 'NMTOKEN' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::NMTOKEN |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::token |

#### A.2.3.32 Datatype NMTOKENS

**Description**

XSD data type 'NMTOKENS' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::NMTOKENS |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::NMTOKEN |

#### A.2.3.33 Datatype nonNegativeInteger

**Description**

XSD data type 'nonNegativeInteger' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::integer |

#### A.2.3.34 Datatype nonPositiveInteger

**Description**

XSD data type 'nonPositiveInteger' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::nonPositiveInteger |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::integer |

#### A.2.3.35 Datatype normalizedString

**Description**

XSD data type 'normalizedString' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::normalizedString |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::string |

#### A.2.3.36 Datatype NOTATION

**Description**

XSD data type 'NOTATION' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::NOTATION |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |

#### A.2.3.37 Datatype positiveInteger

**Description**

XSD data type 'positiveInteger' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::positiveInteger |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |

#### A.2.3.38 Datatype QName

**Description**

XSD data type 'QName' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::QName |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |

#### A.2.3.39 Datatype short

**Description**

XSD data type 'short' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::short |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::int |

#### A.2.3.40 Datatype string

**Description**

XSD data type 'string' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::string |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::String |

#### A.2.3.41 Datatype time

**Description**

XSD data type 'time' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::time |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::anySimpleType |
| Metaclass | ISO20022::Metamodel::DataTypes::Time |

#### A.2.3.42 Datatype token

**Description**

XSD data type 'token' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::token |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::normalizedString |

#### A.2.3.43 Datatype unsignedByte

**Description**

XSD data type 'unsignedByte' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::unsignedByte |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::unsignedShort |

#### A.2.3.44 Datatype unsignedInt

**Description**

XSD data type 'unsignedInt' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::unsignedInt |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::unsignedLong |

#### A.2.3.45 Datatype unsignedLong

**Description**

XSD data type 'unsignedLong' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::unsignedLong |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |

#### A.2.3.46 Datatype unsignedShort

**Description**

XSD data type 'unsignedShort' defined in the XML Schema TypeLibrary.

| | |
|---|---|
| Qualified Name | ISO20022::TypeLibrary::XMLSchema::unsignedShort |
| Visibility | public |
| Supertype | ISO20022::TypeLibrary::XMLSchema::unsignedInt |

# Annex B
(normative)

# Metamodel

## B.1 General

The metamodel for this International Standard is expressed in this annex in MOF.

The metamodel in this part of ISO 20022 formalizes the semantics of the ISO 20022 method. Depictions that are used in other parts of this International Standard may be based on this metamodel. This metamodel however is the definitive reference.

The metamodel uses UML 2.1.2.

## B.2 Metamodel details

### B.2.1 Package ISO20022::Metamodel

#### B.2.1.1 General

| | |
|---|---|
| Applied Stereotype | Metamodel |

#### B.2.1.2 Metaclass BusinessConcept

**Description**

A DataDictionary item with a business meaning.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | |

#### B.2.1.3 Metaclass BusinessProcessCatalogue

**Description**

Part of the ISO 20022 Repository that contains all Business Process and BusinessTransaction related items.

NOTE It contains related items from the BusinessArea down to the MessageDefinitions and their physical implementation.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— repository

**Description**

The Repository that owns the BusinessProcessCatalogue.

| | |
|---|---|
| Type | ISO20022::Metamodel::Repository |
| Owner | ISO20022::Metamodel::BusinessProcessCatalogue |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— TopLevelCatalogueEntry

**Description**

A TopLevelCatalogueEntry in the BusinessProcessCatalogue.

| | |
|---|---|
| Type | ISO20022::Metamodel::TopLevelCatalogueEntry |
| Owner | ISO20022::Metamodel::BusinessProcessCatalogue |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | false |

**Constraints**

— EntriesHaveUniqueName

**Description**

All TopLevelCatalogueEntries of a BusinessProcessCatalogue shall have different names.

| | |
|---|---|
| Context | ISO20022::Metamodel::BusinessProcessCatalogue |
| Language | OCL2.0 |
| Body | TopLevelCatalogueEntry->forAll(entry1,entry2 \| entry1 <> entry2 implies entry1.name <> entry2.name) |

#### B.2.1.4 Metaclass Code

**Description**

A character string (letters, figures or symbols) that for brevity and/or language independence can be used to represent or replace a definitive value or text of an attribute.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— codeName

**Description**

Provides the abbreviated name of the Code.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::Name |
| Owner | ISO20022::Metamodel::Code |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— codeSet

**Description**

The context of the Code.

| | |
|---|---|
| Type | ISO20022::Metamodel::CodeSet |
| Owner | ISO20022::Metamodel::Code |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.1.5 Metaclass CodeSet

**Description**

Set of Codes grouped together to characterize all the values of an attribute.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel |

| | |
|---|---|
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::Text |

**Properties**

— code

**Description**

A set of Codes belonging to the same CodeSet.

| | |
|---|---|
| Type | ISO20022::Metamodel::Code |
| Owner | ISO20022::Metamodel::CodeSet |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | false |

— derivation

**Description**

The formation of a CodeSetTrace from a CodeSet, which is the source.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::CodeSetTrace |
| Owner | ISO20022::Metamodel::CodeSet |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— identificationScheme

**Description**

Uniquely identifies a set of Codes through a Uniform Resource Identifier (URI).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::anyURI |
| Owner | ISO20022::Metamodel::CodeSet |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— trace

**Description**

The formation of a CodeSetTrace from a CodeSet, which is the target.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::CodeSetTrace |
| Owner | ISO20022::Metamodel::CodeSet |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

#### B.2.1.6 Metaclass Constraint

**Description**

Rule that shall be universally satisfied, i.e. all conditions required for the Constraint to be applicable are known.
EXAMPLE An Account shall have an AccountOwner.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— expression

**Description**

The description of a Constraint in a specific language, defined in expressionLanguage.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::Constraint |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— expressionLanguage

**Description**

The language in which a Constraint in expressed.

EXAMPLE OCL 2.0

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::Constraint |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.1.7 Metaclass DataDictionary

**Description**

Part of the ISO 20022 Repository that contains all items that can be re-used during business process modelling and message definition activities.

NOTE The Data Dictionary therefore contains BusinessConcepts, MessageConcepts and DataTypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— repository

**Description**

The Repository that owns the DataDictionary.

| | |
|---|---|
| Type | ISO20022::Metamodel::Repository |
| Owner | ISO20022::Metamodel::DataDictionary |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— topLevelDictionaryEntry

**Description**

A TopLevelDictionaryEntry in the DataDictionary.

| | |
|---|---|
| Type | ISO20022::Metamodel::TopLevelDictionaryEntry |
| Owner | ISO20022::Metamodel::DataDictionary |
| Default Value | |
| Multiplicity | 0..* |

| | |
|---|---|
| Aggregation | composite |
| Ordered | false |

**Constraints**

— EntriesHaveUniqueName

**Description**

All TopLevelDictionaryEntries of a DataDictionary shall have different names.

| | |
|---|---|
| Context | ISO20022::Metamodel::DataDictionary |
| Language | OCL2.0 |
| Body | topLevelDictionaryEntry->forAll(entry1,entry2 \| entry1 <> entry2 implies entry1.name <> entry2.name) |

#### B.2.1.8 Metaclass IdentifierSet

**Description**

Set of values whereby each value distinguishes uniquely one instance of an object within an identification scheme from all other instances of the objects within the same scheme.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::Text |

**Properties**

— identificationScheme

**Description**

Uniquely identifies a set of identifiers through a Uniform Resource Identifier (URI).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::anyURI |
| Owner | ISO20022::Metamodel::IdentifierSet |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.1.9 Metaclass MessageConcept

**Description**

DataDictionary artefact that is used in a MessageDefinition and that is not a DataType.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | |

#### B.2.1.10 Metaclass Repository

**Description**

Place where all RepositoryConcepts are stored.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— businessProcessCatalogue

**Description**

The BusinessProcessCatalogue owned by the ISO 20022 Repository.

| | |
|---|---|
| Type | ISO20022::Metamodel::BusinessProcessCatalogue |
| Owner | ISO20022::Metamodel::Repository |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | composite |
| Ordered | false |

— dataDictionary

**Description**

The DataDictionary owned by the ISO 20022 Repository.

| | |
|---|---|
| Type | ISO20022::Metamodel::DataDictionary |
| Owner | ISO20022::Metamodel::Repository |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | composite |
| Ordered | false |

#### B.2.1.11 Metaclass RepositoryConcept

**Description**

Artefact that has been defined during the development of an ISO 20022 MessageDefinition and which is stored in the Repository.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— constraint

**Description**

A property of a RepositoryConcept specifying a semantic condition or restriction expressed in natural language text and potentially in a formal notation.

EXAMPLE An Account shall have an AccountOwner.

| | |
|---|---|
| Type | ISO20022::Metamodel::Constraint |
| Owner | ISO20022::Metamodel::RepositoryConcept |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— definition

**Description**

Describes the meaning of a RepositoryConcept.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::RepositoryConcept |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— example

**Description**

Provides a representative instance of a RepositoryConcept.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::RepositoryConcept |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— name

**Description**

A word or set of words by which a RepositoryConcept is known, addressed or referred to.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::Name |
| Owner | ISO20022::Metamodel::RepositoryConcept |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— objectIdentifier

**Description**

An ITU-T X.660 | ISO/IEC 9834 series OID (Object Identifier).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::normalizedString |
| Owner | ISO20022::Metamodel::RepositoryConcept |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— registrationStatus

**Description**

Specifies which stage of the registration lifecycle a RepositoryConcept is in.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::RegistrationStatus |
| Owner | ISO20022::Metamodel::RepositoryConcept |
| Default Value | PROVISIONALLY_REGISTERED |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— removalDate

**Description**

Specifies the date at which a RepositoryConcept will cease or has ceased to be part of the Repository.

NOTE When a RepositoryConcept is removed from the Repository, it will be moved to the Repository archive.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::token |
| Owner | ISO20022::Metamodel::RepositoryConcept |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— semanticMarkup

**Description**

Enables modellers to markup elements of the Repository with semantic metadata. Each semanticMarkup string is a TupleValue. A TupleValue is formatted as defined by the following EBNF:

<TupleValue> ::= <TupleTypeName>" : "<TupleElement>[" ,"<TupleElement>]*

<TupleElement> ::= <TupleElementName>" = "<TupleElementValue>

<TupleElementValue> ::= (. - (',' | '=' | ':'))*

In essence, a TupleValue is a string that starts with a name that identifies a TupleType, followed by a colon character, followed by one or more comma-separated TupleElements. A TupleElement is a name-value pair, where the name of the TupleElement and the value of the TupleElement are separated by an equal sign character, with the name of the TupleElement on the left side of the equal sign.

ISO 20022 users can apply to register TupleTypes in the ISO 20022 Repository. A TupleType definition specifies a TupleTypeName and one or more TupleElementNames.

TupleTypeNames, TupleElementNames, and TupleElementValues are case sensitive and may not contain whitespace. They may contain any character that is legal in a UML Tagged Value. However, the colon, comma and equal sign characters are not allowed in TupleElementValues as they are reserved characters required to parse the TupleValues.

There is one predefined TupleType. Its name is "Synonym" and it defines two TupleElements named "name" and "context", as illustrated by the following:

TupleTypeName = Synonym

TupleElementName = name

TupleElementName = context

SemanticMarkup is a multi-valued Property, so it can contain multiple semanticMarkup strings.

EXAMPLE This string shows a TupleValue that is a valid instance of the predefined Synonym TupleType:

Synonym: name = Instrument; context = EuropeanUnion

EXAMPLE A TupleType that could be defined is a reference to an element of an OWL Ontology:

TupleTypeName = OWLResourceId

TupleElementName = URI

EXAMPLE A TupleType that could be defined is a specification of underlying semantic concepts in a manner that promotes alignment with the UN/CEFACT Core Component (CCTS) standard:

TupleTypeName = UnqualifiedBusinessElement

TupleElementName = PropertyTerm

TupleElementName = RepresentationTerm

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::RepositoryConcept |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

**Constraints**

— NameFirstLetterUppercase

**Description**

First letter of name shall be upper case. [A-Z]

| | |
|---|---|
| Context | ISO20022::Metamodel::RepositoryConcept |
| Language | OCL2.0 |
| Body | Set {'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'}->exists(x\|x=name.substring(1,1)) |

— RemovalDateRegistrationStatus

**Description**

If a removalDate is specified then the registrationStatus shall be OBSOLETE

| | |
|---|---|
| Context | ISO20022::Metamodel::RepositoryConcept |
| Language | OCL2.0 |
| Body | removalDate->notEmpty( ) implies registrationStatus = RegistrationStatus::OBSOLETE |

#### B.2.1.12 Metaclass TopLevelCatalogueEntry

**Description**

Artefact in the BusinessProcessCatalogue that is not owned by another artefact in the Repository.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— businessProcessCatalogue

**Description**

The BusinessProcessCatalogue that contains all ISO 20022 TopLevelCatalogueEntries.

| | |
|---|---|
| Type | ISO20022::Metamodel::BusinessProcessCatalogue |
| Owner | ISO20022::Metamodel::TopLevelCatalogueEntry |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.1.13 Metaclass TopLevelDictionaryEntry

**Description**

Artefact in the Dictionary that is not owned by another artefact in the Repository.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— dataDictionary

**Description**

The DataDictionary that contains all ISO 20022 TopLevelDictionaryEntries.

| | |
|---|---|
| Type | ISO20022::Metamodel::DataDictionary |
| Owner | ISO20022::Metamodel::TopLevelDictionaryEntry |
| Default Value | |
| Multiplicity | 1 |

| | |
|---|---|
| Aggregation | none |
| Ordered | false |

#### B.2.1.14 Metaclass Trace

**Description**

Relationship between two objects that represent the same concept but have a different but related context.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel |
| Applied Stereotype | metaclass |
| Superclass | |

**Constraints**

— TraceRules

**Description**

| | |
|---|---|
| Context | ISO20022::Metamodel::Trace |
| Language | English |
| Body | A trace can raise the minimum cardinality and lower the maximum cardinality only. A trace cannot change types. |

#### B.2.1.15 Datatype Cardinality

**Description**

Number of elements in a set.

| | |
|---|---|
| Qualified Name | ISO20022::Metamodel::Cardinality |
| Visibility | public |
| Supertype | |

### B.2.2 Package ISO20022::Metamodel::ConceptualLevel

#### B.2.2.1 General

| | |
|---|---|
| Applied Stereotype | |

### B.2.3 Package ISO20022::Metamodel::ConceptualLevel::Dynamic

#### B.2.3.1 Metaclass BusinessTransaction

**Description**

Particular solution that meets the communication requirements and the interaction requirements of a particular BusinessProcess and BusinessArea.

NOTE It is typically based on the use of MessageInstances.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

**Properties**

— businessProcessTrace

**Description**

The BusinessProcessTrace that is used to trace the BusinessTransaction.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeToConceptualTransformation::BusinessProcessTrace |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageTransportMode

**Description**

Provides a set of characterstics for a MessageTransportMode to have in the context of a single BusinessTransaction.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— parentTransaction

**Description**

Assembly of a number of BusinessTransactions that together form a BusinessTransaction.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Default Value | |

| | |
|---|---|
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— participant

**Description**

The involvement of a BusinessRole in a BusinessTransaction.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | false |

— subTransaction

**Description**

Decomposition of a BusinessTransaction into a number of sub transactions which are BusinessTransactions in their own right.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— trace

**Description**

All of the BusinessTransactionTraces that derive MessageChoreographies from one BusinessTransaction

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessTransactionTrace |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— transmission

**Description**

The conveyance of information from a sending Participant in the context of a BusinessTransaction.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | true |

**Constraints**

— MessageTransmissionsHaveUniqueNames

**Description**

All MessageTransmissions contained in this BusinessTransaction shall have different names.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Language | OCL2.0 |
| Body | transmission->forAll(msg1,msg2 \| msg1 <> msg2 implies msg1.name <> msg2.name) |

— ParticipantsHaveUniqueNames

**Description**

All Participants of this BusinessTransaction shall have different names.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Language | OCL2.0 |
| Body | participant->forAll(p1,p2 \| p1 <> p2 implies p1.name <> p2.name) |

— UniqueName

**Description**

All BusinessTransactions shall have a unique name within the Catalogue.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Language | OCL2.0 |
| Body | topLevelCatalogueEntry -> select(oclIsKindOf(SyntaxMessageScheme)) -> isUnique(name) |

#### B.2.3.2 Metaclass MessageTransmission

**Description**

The passing of information from one Participant to another in the context of a BusinessTransaction.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— businessTransaction

**Description**

The BusinessTransaction to which the MessageTransmission belongs.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— derivation

**Description**

All of the MessagetypeTraces that derive MessageDefinitions from one MessageTransmission.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::MessageTypeTrace |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— messageTypeDescription

**Description**

Describes the purpose and scope of the MessageTransmission in the BusinessTransaction.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |

| | |
|---|---|
| Ordered | false |

— receive

**Description**

The handling of a MessageTransmission passed from a sender instance.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::Receive |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | false |

— send

**Description**

The passing of a MessageTransmission from a sender instance to a receiver instance.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::Send |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | composite |
| Ordered | false |

#### B.2.3.3 Metaclass MessageTransportMode

**Description**

Group of settings for the values for the MessageTransportCharacteristics properties.

NOTE 1 A MessageTransportMode is named and registered in the ISO 20022 Repository. Each MessageTransportCharacteristic is given a value.

NOTE 2 A MessageTransportMode can be associated with many BusinessTransactions. The MessageTransportMode is used to organize commonly used combinations of MessageTransportCharacteristic settings.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

**Properties**

— boundedCommunicationDelay

**Description**

The maximum duration of time within which a TransportMessage shall be delivered.

NOTE 3 The valid value of this property is any positive ISO 8601 duration greater than zero.

NOTE 4 A TransportMessage that is not delivered within the BoundedCommunicationDelay cannot be delivered. A receiving MessagingEndpoint may ignore a TransportMessage that is delivered outside of the BoundedCommunicationDelay. Whether a notification of an error occurs when the BoundedCommunicationDelay is exceeded is determined by the DeliveryAssurance setting.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::duration |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— businessTransaction

**Description**

Specifies the BusinessTransaction for which a set of MessageTransportMode characteristics apply.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— deliveryAssurance

**Description**

The degree to which the sending MessagingEndpoint is assured that a TransportMessage will be delivered.

NOTE 5 The MessageTransportSystem is responsible for implementing this characteristic and may therefore decide to achieve this by republishing TransportMessages. The Idempotent Behaviours apply.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::DeliveryAssurance |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— durability

**Description**

Whether the MessageTransportSystem safely retains a TransportMessage until it has been received by the destination MessagingEndpoint.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::Durability |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— maximumClockVariation

**Description**

Clocks shall maintain a maximum (inclusive) variance from UTC for the supported MessageTransportMode.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::duration |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— maximumMessageSize

**Description**

The maximum size of a TransportMessage in kilobytes (any positive integer greater than zero).

NOTE 6 Any TransportMessage exceeding the MaximumMessageSize is treated as an invalid message. The specified behaviour for invalid messages applies.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::positiveInteger |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageCasting

**Description**

Specifies how receiving MessagingEndpoints may be addressed in a TransportMessage.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::MessageCasting |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageDeliveryOrder

**Description**

Indicates to what extent Transport Messages from a sending MessagingEndpoint arrive in the order in which they were sent at the receiving MessagingEndpoints.

NOTE 7 The sending order is defined as the order in which the messages arrive at the MessageTransportSystem.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::MessageDeliveryOrder |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageDeliveryWindow

**Description**

The MessageDeliveryOrder is relaxed by a rolling window of time in which the Transport Messages may be delivered out of order, whereby for each TransportMessage delivered it defines the maximum duration of time within which it may be out of sequence.

NOTE 8 A relaxation of the MessageDeliveryOrder has no effect on a MessageDeliveryOrder of UNORDERED because the order cannot be relaxed further.

NOTE 9 A MessageDeliveryWindow of zero duration is equivalent to no window, i.e. a strict ordering of delivery.

NOTE 10 The purpose of this is to enable implementers of a MessageTransportSystem to implement a windowing mechanism to reorder messages slightly out of order.

NOTE 11 This is a relaxation of ordering of delivery at the MessageTransportLayer.

NOTE 12 Values are greater than or equal to zero.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::duration |

| | |
|---|---|
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageSendingWindow

**Description**

The Choreography is relaxed by a rolling window of time in which the Business Messages may be sent out of order, whereby for each Business Message sent it defines the maximum duration of time within which it may be out of sequence.

NOTE 13 It is not possible to send messages unordered as this would contradict the Choreography.

NOTE 14 The purpose of the MessageSendingWindow is to prevent bottlenecks on sending messages by reducing the need to coordinate sending.

NOTE 15 This is a relaxation of ordering of sending at the Business Layer.

NOTE 16 Values are greater than or equal to zero.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::duration |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageValidationLevel

**Description**

The level of validation to which the MessageTransportSystem has tested the message.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::MessageValidationLevel |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageValidationOnOff

**Description**

Specifies whether the MessageTransportSystem validates the MessageInstance with respect to SyntaxMessageScheme, Constraints, the MarketPractices, and the MessageChoreography.

NOTE 17 The validation occurs before the MessageInstance is delivered to the receiving MessagingEndpoints.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::MessageValidationOnOff |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageValidationResults

**Description**

Specifies how the MessageTransport System acts upon the results of MessageValidationOnOff.

NOTE 18 This characteristic only applies when MessageValidationOnOff characteristic has the value of VALIDATION_ON; if the MessageValidationOnOff is VALIDATION_OFF then there is to be no record of the validation results and this characteristic has no effect.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::MessageValidationResults |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— receiverAsynchronicity

**Description**

Indicates whether a receiving MessagingEndpoint blocks the sending and receipt of other Transport Messages until it sends a response to this TransportMessage.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::ReceiverAsynchronicity |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— senderAsynchronicity

**Description**

Indicates whether a sending MessagingEndpoint blocks after sending a TransportMessage to the MessageTransportSystem while waiting for a response from a MessagingEndpoint.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::SenderAsynchronicity |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransportMode |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.3.4 Metaclass Participant

**Description**

Involvement of a BusinessRole in a BusinessTransaction.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— businessRoleTrace

**Description**

The semantic trace between the Participant and its BusinessRole.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeToConceptualTransformation::BusinessRoleTrace |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— businessTransaction

**Description**

The business transaction in which the Participant plays a role.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |

| | |
|---|---|
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— cardinality

**Description**

The number of Participants that can be the recipient of a MessageTransmission.

EXAMPLE A BroadCast is typically sent to more than one Participant.

| | |
|---|---|
| Type | ISO20022::Metamodel::Cardinality |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— receives

**Description**

The handling of a stimulus passed from a sending participant.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::Receive |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— sends

**Description**

The passing of information from a sending Participant to a receiving Participant.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::Send |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

#### B.2.3.5 Metaclass Receive

**Description**

The handling of a stimulus passed from a sender instance.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— messageTransmission

**Description**

The stimulus handled as part of the Receive.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Receive |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— receiver

**Description**

The object handling a stimulus passed from the sender.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Receive |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.3.6 Metaclass Send

**Description**

The passing of a stimulus from a sender instance to a receiver instance.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic |

| | |
|---|---|
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— messageTransmission

**Description**

The stimulus handled as part of the Send.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Send |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— sender

**Description**

The object passing a stimulus to a receiver object.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |
| Owner | ISO20022::Metamodel::ConceptualLevel::Dynamic::Send |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

## B.2.4 Package ISO20022::Metamodel::ConceptualLevel::MessageTransport

### B.2.4.1 Metaclass Address

**Description**

Identification and efficient resolution to the location of a MessagingEndpoint.

NOTE The purpose of an Address is to efficiently resolve a location. This is what distinguishes an Address from any other Identifier, which merely identifies something.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— broadCastList

**Description**

A BroadCastList to which this Address belongs.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::BroadcastList |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::Address |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— endpoint

**Description**

Specifies the MessagingEndpoint for the Address.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessagingEndpoint |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::Address |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

### B.2.4.2 Metaclass BroadcastList

**Description**

Set of references to MessagingEndpoints (identified by their Address), which is used for message broadcasting.

NOTE 1 The BroadcastList is managed by the MessageTransportSystem, which provides a mechanism to maintain the BroadcastList.

NOTE 2 “Set” means the list of Addresses is unordered and each Address can only be present once.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— address

**Description**

A BroadCastList entry using an Address.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::Address |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::BroadcastList |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | shared |
| Ordered | false |

### B.2.4.3 Metaclass Conversation

**Description**

Exchange of one or more MessageInstances among MessagingEndpoints.

NOTE In a synchronous Conversation, the sending MessagingEndpoint blocks the sending and receipt of other TransportMessages within the conversation, in which the TransportMessage was sent, while waiting for a response to this sent TransportMessage. This is not the case in an asynchronous conversation.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport |
| Applied Stereotype | metaclass |
| Superclass | |

### B.2.4.4 Metaclass MessageTransportSystem

**Description**

Mechanism that receives Transport Messages from the sending MessagingEndpoint, transports them, and delivers them to the receiving MessagingEndpoint.

NOTE 1 The MessageTransportSystem is responsible for delivering Transport Messages to each Addressee.

NOTE 2 The purpose of the MessageTransportSystem is to provide a clear delineation of the responsibility of the MessagingEndpoints and any MessageTransportSystem service providers. The role can be fulfilled by the sending MessagingEndpoint or by a separate service provider who provides a MessageTransportSystem. MessagingTransportSystems can be chained together into a single MessageTransportSystem.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— endpoint

**Description**

A MessagingEndpoint owned by a single MessageTransportSystem.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessagingEndpoint |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessageTransportSystem |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | false |

### B.2.4.5 Metaclass MessagingEndpoint

**Description**

Addressable node on the MessageTransportSystem which is capable of sending and receiving TransportMessages.
NOTE A MessagingEndpoint has an Address.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— location

**Description**

An Address used to identify the MessagingEndpoint.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::Address |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessagingEndpoint |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— receivedMessage

**Description**

The TransportMessage that is received by the receiving MessagingEndpoint.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::TransportM |

| | |
|---|---|
| | essage |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessagingEndpoint |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— sentMessage

**Description**

The TransportMessage that is sent by the sending MessagingEndpoint.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::TransportMessage |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessagingEndpoint |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— transportSystem

**Description**

The MessageTransportSystem that owns and uses this MessagingEndpoint.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessageTransportSystem |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessagingEndpoint |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.4.6 Metaclass TransportMessage

**Description**

Document that is an instance of the MessageTransportSystem message schema.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport |
| Applied Stereotype | metaclass |

| Superclass | |
|---|---|

**Properties**

— messageInstance

**Description**

The MessageInstance that is part of the TransportMessage.

| Type | ISO20022::Metamodel::PhysicalLevel::MessageInstance |
|---|---|
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::TransportMessage |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— receiver

**Description**

The receiving MessagingEndpoint in a TransportMessage.

| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessagingEndpoint |
|---|---|
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::TransportMessage |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— sender

**Description**

The sending MessagingEndpoint of a TransportMessage.

| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::MessagingEndpoint |
|---|---|
| Owner | ISO20022::Metamodel::ConceptualLevel::MessageTransport::TransportMessage |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— sameMessageTransportSystem

**Description**

The sender and receiver of a TransportMessage shall use the same MessageTransportSystem.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::MessageTransport::TransportMessage |
| Language | OCL2.0 |
| Body | this.sender.transportSystem = this.receiver.transportSystem |

## B.2.5 Package ISO20022::Metamodel::ConceptualLevel::Static

### B.2.5.1 Metaclass BusinessAssociation

**Description**

Relation between two BusinessComponents.

EXAMPLE A party services an account.

NOTE 1 BusinessAssociations are a category of BusinessConcepts. Their meaning can only be described unambiguously in combination with these two BusinessComponents.

NOTE 2 There can be several semantic relations between two particular BusinessComponents.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::BusinessConcept ISO20022::Metamodel::TopLevelDictionaryEntry |

**Properties**

— endA

**Description**

Represents one of the two BusinessAssociationEnds connecting a BusinessAssociation to a BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociationEnd |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociation |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | composite |
| Ordered | false |

— endB

**Description**

Represents one of the two BusinessAssociationEnds connecting a BusinessAssociation to a BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociationEnd |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociation |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | composite |
| Ordered | false |

**Constraints**

— AtMostOneAggregatedEnd

**Description**

The two association ends may not have composite or shared aggregation at the same time.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociation |
| Language | OCL2.0 |
| Body | not(endA.aggregation <> Aggregation.NONE and endB.aggregation <> Aggregation.NONE) |

— ContextConsistentWithType

**Description**

The context of endB shall be the type of endA, and vice-versa.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociation |
| Language | OCL2.0 |
| Body | endB.context = endA.type and endA.context = endB.type |

— UniqueName

**Description**

All BusinessAssociations shall have a unique name within the Dictionary.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociation |
| Language | OCL2.0 |
| Body | topLevelDictionaryEntry -> select(oclIsKindOf(BusinessAssociation)) -> isUnique(name) |

#### B.2.5.2 Metaclass BusinessAssociationEnd

**Description**

The endpoint of a BusinessAssociation, which connects the BusinessAssociation to the BusinessComponent.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::ConceptualLevel::Static::BusinessElement |

**Properties**

— aggregation

**Description**

Expresses the strength of the semantic relationship between two BusinessComponents.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::Aggregation |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociationEnd |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— associationForEndA

**Description**

The association for one of the two AssociationEnds, more specifically EndA.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociation |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociationEnd |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— associationForEndB

**Description**

The association for one of the two AssociationEnds, more specifically EndB.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociation |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociationEnd |
| Default Value | |

| | |
|---|---|
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— type

**Description**

Specifies that a BusinessAssociationEnd always has a complex content model and is therefore always typed by a BusinessComponent, in contrast to a BusinessAttribute, which may be typed by a data type.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociationEnd |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— EitherAnEndAOrAnEndB

**Description**

A BusinesssAssociationEnd shall play exactly one of the two following roles: endA or endB.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociationEnd |
| Language | OCL2.0 |
| Body | associationForEndA->isEmpty() xor associationForEndB->isEmpty() |

### B.2.5.3 Metaclass BusinessAttribute

**Description**

A BusinessElement, typed by a BusinessComponent or a DataType (contrary to a BusinessAssociationEnd, which is always typed by another BusinessComponent).

EXAMPLE AccountIdentification, PhoneNumber.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::ConceptualLevel::Static::BusinessElement ISO20022::Metamodel::RepositoryConcept |

**Properties**

— complexType

**Description**

The BusinessComponent that describes the complex content model of the BusinessAttrribute.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAttribute |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— simpleType

**Description**

Expresses that the content model of a BusinessAttribute may be specified by a type from the XSD type library or a derived datatype.

| | |
|---|---|
| Type | ISO20022::Metamodel::DataTypes::DataType |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAttribute |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— BusinessAttributeHasExactlyOneType

**Description**

A BusinessAttribute shall have exactly one of the following: simpleType and complexType.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAttribute |
| Language | OCL2.0 |
| Body | complexType->size() + simpleType->size() = 1 |

— NoDerivingCodeSetType

**Description**

Deriving Code Sets may only be used to type MessageAttributes. Therefore, a BusinessAttribute may not be typed by a Deriving CodeSet.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAttribute |
| Language | OCL2.0 |

| | |
|---|---|
| Body | type.oclIsKindOf(CodeSet) implies type.oclAsType(CodeSet).trace->isEmpty() |

#### B.2.5.4 Metaclass BusinessComponent

**Description**

Representation of a (part of a) key business notion, characterized by specific BusinessElements.

EXAMPLE Account, trade, party.

NOTE 1 BusinessComponents are a category of BusinessConcepts. They are stored in the DataDictionary.

NOTE 2 A BusinessComponent can have one or more BusinessAssociations with other BusinessComponents.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::BusinessConcept<br>ISO20022::Metamodel::TopLevelDictionaryEntry |

**Properties**

— associationDomain

**Description**

Describes the semantics that determine how the BusinessComponent may participate in the BusinessAssociation.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessAssociationEnd |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— derivation

**Description**

All of the BusinessComponentTraces that derive MessageComponentTypes and MessageElements from one BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessComponentTrace |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Default Value | |
| Multiplicity | 0..* |

| | |
|---|---|
| Aggregation | none |
| Ordered | false |

— element

**Description**

A semantic property of a BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessElement |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | true |

— subType

**Description**

The BusinessComponents that specialize this BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Default Value | |
| Multiplicity | * |
| Aggregation | none |
| Ordered | false |

— superType

**Description**

The BusinessComponent that is specialized by this BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— BusinessElementsHaveUniqueNames

**Description**

All BusinessElements contained by this BusinessComponents have different names.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Language | OCL2.0 |
| Body | element->forAll(el1,el2 : BusinessElement\| el1 <> el2 implies el1.name <> el2.name) |

— UniqueName

**Description**

All BusinessComponents shall have a unique name within the Dictionary.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Language | OCL2.0 |
| Body | topLevelDictionaryEntry -> select(oclIsKindOf(SyntaxMessageScheme)) -> isUnique(name) |

#### B.2.5.5 Metaclass BusinessElement

**Description**

Property of a BusinessComponent that has a distinctive meaning within the scope of that BusinessComponent.

EXAMPLE Account status, deal price, trade date and deal time.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::BusinessConcept ISO20022::Metamodel::Repository Concept |

**Properties**

— cardinality

**Description**

Property of a BusinessElement describing the number of allowed occurrences it may have in a particular BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::Cardinality |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessElement |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— derivation

**Description**

All of the BusinessElementTraces that derive MessageElements from one BusinessElement in a specific BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessElementTrace |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessElement |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— elementContext

**Description**

The business context in which the BusinessElement is used.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessElement |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— isDerived

**Description**

Specifies whether a BusinessElement can be computed using other BusinessElements. It is shown for clarity even though it adds no semantic information.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::boolean |
| Owner | ISO20022::Metamodel::ConceptualLevel::Static::BusinessElement |
| Default Value | false |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

## B.2.6 Package ISO20022::Metamodel::ConceptualToLogicalTransformation

### B.2.6.1 Metaclass BusinessComponentTrace

**Description**

Semantic relationship between a MessageComponentType/MessageElement and the BusinessComponent from which it is derived.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::Trace |

**Properties**

— businessComponent

**Description**

The BusinessComponent to which a MessageComponentType or MessageElement is traced.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessComponent |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessComponentTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageComponentType

**Description**

The MessageComponentType that is traced to a BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessComponentTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageElement

**Description**

The MessageElement that is traced to a BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessComponentTrace |
| Default Value | |
| Multiplicity | 1 |

| | |
|---|---|
| Aggregation | none |
| Ordered | false |

#### B.2.6.2 Metaclass BusinessElementTrace

**Description**

Semantic relationship between a MessageElement and the BusinessElement from which it is derived.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::Trace |

**Properties**

— businessElement

**Description**

The BusinessElement to which a MessageElement is traced.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Static::BusinessElement |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessElementTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageElement

**Description**

The MessageElement that is traced to a BusinessElement.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessElementTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— CardinalityAlignment

**Description**

A trace can only raise the minimum cardinality and can only lower the maximum cardinality.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessElementTrace |
| Language | OCL2.0 |
| Body | toReal(cardinality.minimumOccurrence) >= toReal(businessElement.cardinality.minimumOccurrence) and toReal(cardinality.maximumOccurrence) <= toReal(businessElement.cardinality.maximumOccurrence) |

— MaxGEzero

**Description**

MaximumOccurrence shall be greater than 0.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessElementTrace |
| Language | OCL2.0 |
| Body | maximumOccurrence > 0 |

— MinLEmax

**Description**

MinimumOccurrence should not exceed maximumOccurrence.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessElementTrace |
| Language | OCL2.0 |
| Body | minimumOccurrence->notEmpty( ) and maximumOccurrence->notEmpty( ) implies minimumOccurrence <= maximumOccurrence |

— SameRepresentation

**Description**

When the businessElement is a BusinessAttribute, then the messageElement shall be a MessageAttribute with the same kind of type as the BusinessAttribute.

| | |
|---|---|
| Context | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessElementTrace |
| Language | OCL2.0 |
| Body | (businessElement.oclIsKindOf(BusinessAttribute) and not businessElement.simpleType.oclIsUndefined() |

| | |
|---|---|
| | and messageElement.oclIsKindOf(MessageAttribute)<br>and<br>businessElement.simpleType.oclType.isCompatibleWith(messageElement<br>.simpleType.oclType)<br>)<br>or<br>(businessElement.oclIsKindOf(BusinessAttribute)<br>and not businessElement.complexType.oclIsUndefined()<br>and messageElement.oclIsKindOf(MessageAttribute)<br>and messageElement.oclIsKindOf(MessageComponentType)<br>) |

#### B.2.6.3 Metaclass BusinessTransactionTrace

**Description**

Relationship between a BusinessTransaction and its physical implementation.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::Trace |

**Properties**

— businessTransaction

**Description**

The BusinessTransaction to which the MessageChoreography is traced.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessTransactionTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageChoreography

**Description**

The MessageChoreography that is traced to the BusinessTransaction.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageChoreography |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessTransactionTrace |

| | |
|---|---|
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.6.4 Metaclass CodeSetTrace

**Description**

Semantic relationship between two CodeSets whereby the derived Codeset is used as the type of a BusinessElement and the deriving Codeset is used as the type of a MessageElement.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::Trace |

**Properties**

— derivedCodeSet

**Description**

The CodeSet that is a subset of another CodeSet to which it is traced.

| | |
|---|---|
| Type | ISO20022::Metamodel::CodeSet |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::CodeSetTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— sourceCodeSet

**Description**

The CodeSet from which another CodeSet is derived.

| | |
|---|---|
| Type | ISO20022::Metamodel::CodeSet |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::CodeSetTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.6.5 Metaclass MessageTypeTrace

**Description**

Relationship between a MessageTransmission in a BusinessTransaction and its corresponding MessageDefinition.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::Trace |

**Properties**

— messageDefinition

**Description**

The MessageDefinition that is traced to a MessageTransmission.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::MessageTypeTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageTransmission

**Description**

The MessageTransmission to which a MessageDefinition is traced.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::MessageTransmission |
| Owner | ISO20022::Metamodel::ConceptualToLogicalTransformation::MessageTypeTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

### B.2.7 Package ISO20022::Metamodel::DataTypes

#### B.2.7.1 Metaclass Amount

**Description**

A number of monetary units specified in a currency where the unit of currency is explicit or implied.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::Decimal |

**Properties**

— currencyIdentifierSet

**Description**

Specifies the allowed currencies that can be used to qualify this amount.

| | |
|---|---|
| Type | ISO20022::Metamodel::IdentifierSet |
| Owner | ISO20022::Metamodel::DataTypes::Amount |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

### B.2.7.2 Metaclass Binary

**Description**

Any set of values drawn from the value space of 'base64Binary', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— length

**Description**

The number of units of characters.

NOTE 1 maxLength is always greater than or equal to minLength.

NOTE 2 A unit in this case is a finite-length sequence of binary octets.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |
| Owner | ISO20022::Metamodel::DataTypes::Binary |
| Default Value | |
| Multiplicity | 0..1 |

| | |
|---|---|
| Aggregation | none |
| Ordered | false |

— maxLength

**Description**

The maximum number of units of characters.

NOTE 3 maxLength is always greater than or equal to minLength.

NOTE 4 A unit in this case is a finite-length sequence of binary octets.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |
| Owner | ISO20022::Metamodel::DataTypes::Binary |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minLength

**Description**

The minimum number of units of characters.

NOTE 5 minlength is always smaller than or equal to maxLength.

NOTE 6 A unit in this case is a finite-length sequence of binary octets.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |
| Owner | ISO20022::Metamodel::DataTypes::Binary |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Binary |
| Default Value | |
| Multiplicity | 0..1 |

| | |
|---|---|
| Aggregation | none |
| Ordered | false |

#### B.2.7.3 Metaclass Boolean

**Description**

Any set of values drawn from the value space of 'boolean', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— pattern

**Description**

A constraint on the value space of a datatype which is achieved by constraining the lexical space to literals which match a specific pattern.

NOTE The value of a pattern is a regular expression.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Boolean |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.4 Metaclass DataType

**Description**

Representation of a set of values without identity.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelDictionaryEntry |

#### B.2.7.5 Metaclass Date

**Description**

Any set of values drawn from the value space of 'date', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::date |
| Owner | ISO20022::Metamodel::DataTypes::Date |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::date |
| Owner | ISO20022::Metamodel::DataTypes::Date |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::date |
| Owner | ISO20022::Metamodel::DataTypes::Date |

| | |
|---|---|
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::date |
| Owner | ISO20022::Metamodel::DataTypes::Date |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Date |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.6 Metaclass DateTime

**Description**

Any set of values drawn from the value space of 'dateTime', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::dateTime |
| Owner | ISO20022::Metamodel::DataTypes::DateTime |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::dateTime |
| Owner | ISO20022::Metamodel::DataTypes::DateTime |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::dateTime |
| Owner | ISO20022::Metamodel::DataTypes::DateTime |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::dateTime |
| Owner | ISO20022::Metamodel::DataTypes::DateTime |
| Default Value | |

| | |
|---|---|
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::DateTime |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

### B.2.7.7 Metaclass Day

**Description**

Any set of values drawn from the value space of 'gDay', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gDay |
| Owner | ISO20022::Metamodel::DataTypes::Day |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gDay |
| Owner | ISO20022::Metamodel::DataTypes::Day |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gDay |
| Owner | ISO20022::Metamodel::DataTypes::Day |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gDay |
| Owner | ISO20022::Metamodel::DataTypes::Day |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Day |

| | |
|---|---|
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.8 Metaclass Decimal

**Description**

Any set of values drawn from the value space of 'decimal', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— fractionDigits

**Description**

The fractional part of a Decimal number.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |
| Owner | ISO20022::Metamodel::DataTypes::Decimal |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::decimal |
| Owner | ISO20022::Metamodel::DataTypes::Decimal |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::decimal |
| Owner | ISO20022::Metamodel::DataTypes::Decimal |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::decimal |
| Owner | ISO20022::Metamodel::DataTypes::Decimal |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::decimal |
| Owner | ISO20022::Metamodel::DataTypes::Decimal |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype which is achieved by constraining the lexical space to literals which match a specific pattern.

NOTE The value of a pattern is a regular expression.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Decimal |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— totalDigits

**Description**

The maximum number of allowed digits in a Decimal number

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |
| Owner | ISO20022::Metamodel::DataTypes::Decimal |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

### B.2.7.9 Metaclass Duration

**Description**

Any set of values drawn from the value space of 'duration', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::duration |
| Owner | ISO20022::Metamodel::DataTypes::Duration |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |

| | |
|---|---|
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::duration |
| Owner | ISO20022::Metamodel::DataTypes::Duration |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::duration |
| Owner | ISO20022::Metamodel::DataTypes::Duration |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::duration |
| Owner | ISO20022::Metamodel::DataTypes::Duration |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Duration |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.10 Metaclass Indicator

**Description**

A list of exactly two mutually exclusive values that express the only two possible states of an instance of an object.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::Boolean |

**Properties**

— meaningWhenFalse

**Description**

Provides the semantic meaning when the Indicator is set to false.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Indicator |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— meaningWhenTrue

**Description**

Provides the semantic meaning when the Indicator is set to true.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Indicator |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.11 Metaclass Month

**Description**

Any set of values drawn from the value space of 'gMonth', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gMonth |
| Owner | ISO20022::Metamodel::DataTypes::Month |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gMonth |
| Owner | ISO20022::Metamodel::DataTypes::Month |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gMonth |
| Owner | ISO20022::Metamodel::DataTypes::Month |

| | |
|---|---|
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gMonth |
| Owner | ISO20022::Metamodel::DataTypes::Month |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Month |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.12 Metaclass MonthDay

**Description**

Any set of values drawn from the value space of 'gMonthDay', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gMonthDay |
| Owner | ISO20022::Metamodel::DataTypes::MonthDay |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gMonthDay |
| Owner | ISO20022::Metamodel::DataTypes::MonthDay |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gMonthDay |
| Owner | ISO20022::Metamodel::DataTypes::MonthDay |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gMonthDay |

| | |
|---|---|
| Owner | ISO20022::Metamodel::DataTypes::MonthDay |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::MonthDay |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

### B.2.7.13 Metaclass Quantity

**Description**

A counted number of non-monetary units, possibly including fractions.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::Decimal |

**Properties**

— unitCode

**Description**

Qualifies the value of a Quantity.

EXAMPLE kg

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Quantity |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

### B.2.7.14 Metaclass Rate

**Description**

A quantity or amount measured with respect to another measured quantity or amount.

EXAMPLE 1 US Dollars per hour, US Dollars per EURO.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::Decimal |

**Properties**

— baseUnitCode

**Description**

Specifies unit code required to qualify this rate.

EXAMPLE 2 m/s

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::token |
| Owner | ISO20022::Metamodel::DataTypes::Rate |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— baseValue

**Description**

Specifies the ratio between the base value and the actual value.

EXAMPLE 3 In case of a PercentageRate, the baseValue would be 100.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::decimal |
| Owner | ISO20022::Metamodel::DataTypes::Rate |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.15 Metaclass String

**Description**

Any set of values drawn from the value space of 'string', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— length

**Description**

The number of units of characters (always greater than zero).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |
| Owner | ISO20022::Metamodel::DataTypes::String |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxLength

**Description**

The number of units of characters (always greater than zero and minLength).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::nonNegativeInteger |
| Owner | ISO20022::Metamodel::DataTypes::String |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minLength

**Description**

The minimum number of units of characters (always smaller than or equal to maxLength).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::String |

| | |
|---|---|
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::String |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.16 Metaclass Text

**Description**

A finite set of characters.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::String |

#### B.2.7.17 Metaclass Time

**Description**

Any set of values drawn from the value space of 'time', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::time |
| Owner | ISO20022::Metamodel::DataTypes::Time |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::time |
| Owner | ISO20022::Metamodel::DataTypes::Time |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::time |
| Owner | ISO20022::Metamodel::DataTypes::Time |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::time |
| Owner | ISO20022::Metamodel::DataTypes::Time |
| Default Value | |

| | |
|---|---|
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Time |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

### B.2.7.18 Metaclass Year

**Description**

Any set of values drawn from the value space of 'gYear', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gYear |
| Owner | ISO20022::Metamodel::DataTypes::Year |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gYear |
| Owner | ISO20022::Metamodel::DataTypes::Year |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gYear |
| Owner | ISO20022::Metamodel::DataTypes::Year |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gYear |
| Owner | ISO20022::Metamodel::DataTypes::Year |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::Year |
| Default Value | |

| | |
|---|---|
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

#### B.2.7.19 Metaclass YearMonth

**Description**

Any set of values drawn from the value space of 'gYearMonth', as specified by W3C Recommendation XML Schema Part 2: Datatypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::DataTypes |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::DataTypes::DataType |

**Properties**

— maxExclusive

**Description**

The highest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gYearMonth |
| Owner | ISO20022::Metamodel::DataTypes::YearMonth |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— maxInclusive

**Description**

The highest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gYearMonth |
| Owner | ISO20022::Metamodel::DataTypes::YearMonth |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minExclusive

**Description**

The lowest but one value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gYearMonth |
| Owner | ISO20022::Metamodel::DataTypes::YearMonth |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— minInclusive

**Description**

The lowest value in the allowed set of values.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::gYearMonth |
| Owner | ISO20022::Metamodel::DataTypes::YearMonth |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— pattern

**Description**

A constraint on the value space of a datatype, which is achieved by constraining the lexical space to literals that match a specific pattern (the value of a pattern shall be a regular expression).

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::DataTypes::YearMonth |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

## B.2.8 Package ISO20022::Metamodel::LogicalLevel

### B.2.8.1 Metaclass BusinessArea

**Description**

Set of strongly related business activities that provide a self-standing business value to a set of BusinessRoles.

EXAMPLE Securities pre-trade, payment initiation.

NOTE BusinessAreas are stored in the BusinessProcessCatalogue.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

**Properties**

— code

**Description**

The value of the BusinessArea code.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::NMTOKEN |
| Owner | ISO20022::Metamodel::LogicalLevel::BusinessArea |
| Default Value | |
| Multiplicity | |
| Aggregation | none |
| Ordered | false |

— messageDefinition

**Description**

The MessageDefinition that belongs to the BusinessArea.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Owner | ISO20022::Metamodel::LogicalLevel::BusinessArea |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | false |

**Constraints**

— UniqueName

**Description**

All BusinessAreas shall have a unique name within the Catalogue.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::BusinessArea |
| Language | OCL2.0 |
| Body | topLevelCatalogueEntry -> select(oclIsKindOf(SyntaxMessageScheme)) -> isUnique(name) |

#### B.2.8.2 Metaclass ChoiceComponent

**Description**

Re-usable Dictionary Item composed of a choice of MessageElements, which is a building block for assembling MessageDefinitions.

NOTE 1 It is usually linked to a BusinessComponent.

NOTE 2 ChoiceComponents are stored in the DataDictionary.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::LogicalLevel::MessageComponentType |

**Constraints**

— AtLeastOneProperty

**Description**

A ChoiceComponent shall have at least one MessageElement.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::ChoiceComponent |
| Language | OCL2.0 |
| Body | messageElement->notEmpty( ) |

#### B.2.8.3 Metaclass Encoding

**Description**

A particular set of encoding rules for a MessageInstance.

EXAMPLE 1 ASN.1 PER

EXAMPLE 2 ASN.1 XER

EXAMPLE 3 XSD XML

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— messageSet

**Description**

The MessageSets for which this Encoding is a valid ISO 20022 encoding.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageSet |
| Owner | ISO20022::Metamodel::LogicalLevel::Encoding |
| Default Value | |
| Multiplicity | * |
| Aggregation | none |
| Ordered | false |

— syntax

**Description**

The source syntax on which an encoding applies.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::Syntax |
| Owner | ISO20022::Metamodel::LogicalLevel::Encoding |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.8.4 Metaclass ExternalSchema

**Description**

Reusable Dictionary Item that allows referral to a structure defined outside the ISO 20022 MessageDefinition.

EXAMPLE In case of XML, this artefact is transformed into an XML Schema "any" element and the external structure is defined through another XML Schema.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::LogicalLevel::MessageComponentType |

— namespaceList

**Description**

Identifies the description of the content model of an ExternalSchema, through (a set of) URI.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::LogicalLevel::ExternalSchema |
| Default Value | |

| | |
|---|---|
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— processContent

**Description**

Specifies whether it is necessary to validate the content model of the ExternalSchema.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::Enumerations::ProcessContent |
| Owner | ISO20022::Metamodel::LogicalLevel::ExternalSchema |
| Default Value | |
| Multiplicity | |
| Aggregation | none |
| Ordered | false |

**Constraints**

— NoMessageElement

**Description**

An ExternalSchema may not contain MessageElements, as it is an open content element.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::ExternalSchema |
| Language | OCL2.0 |
| Body | messageElement->isEmpty() |

### B.2.8.5 Metaclass MessageAssociationEnd

**Description**

Kind of MessageElement that relates two MessageComponentTypes.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::LogicalLevel::MessageElement |

**Properties**

— isComposite

**Description**

Specifies the kind of relationship between two MessageComponentTypes.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::string |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageAssociationEnd |

| | |
|---|---|
| Default Value | |
| Multiplicity | 1 |
| Aggregation | composite |
| Ordered | false |

**Properties**

— type

**Description**

The MessageComponentType that specifies the complex content model of a MessageAssociationEnd.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageAssociationEnd |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— OppositesNotBotAggregate

**Description**

If the value of a MessageAssociationEnd's "aggregation" Property is anything other than Aggregation::none, then the value of its "opposite" MessageAssociationEnd shall be Aggregation::none.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageAssociationEnd |
| Language | |
| Body | self.aggregation <> Aggregation::none implies opposite.aggregation = Aggregation::none |

### B.2.8.6 Metaclass MessageAttribute

**Description**

MessageElement, the type of which is either DataType or MessageComponentType.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::LogicalLevel::MessageElement |

**Properties**

— complexType

**Description**

The complex content model of a MessageAttribute when it is expressed using a MessageComponentType.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageAttribute |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— simpleType

**Description**

The simple content model of a MessageAttribute when it is expressed using a DataType.

| | |
|---|---|
| Type | ISO20022::Metamodel::DataTypes::DataType |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageAttribute |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— MessageAttributeHasExactlyOneType

**Description**

A MessageAttribute shall have exactly one of each of the following: simpleType and complexType.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageAttribute |
| Language | OCL2.0 |
| Body | complexType->size() + simpleType->size() = 1 |

### B.2.8.7 Metaclass MessageBuildingBlock

**Description**

Characteristic of a MessageDefinition, which has a unique meaning within the scope of that MessageDefinition.

NOTE MessageBuildingBlocks are not reused, since they only have meaning within a specific MessageDefinition.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— cardinality

**Description**

Property of a MessageBuildingBlock describing the number of allowed occurrences it may have in a particular MessageDefinition.

| | |
|---|---|
| Type | ISO20022::Metamodel::Cardinality |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageBuildingBlock |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— complexType

**Description**

The complex content model of a MessageBuildingBlock when it is expressed using a MessageComponentType.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageBuildingBlock |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— simpleType

**Description**

The simple content model of a MessageBuildingBlock when it is expressed using a DataType.

| | |
|---|---|
| Type | ISO20022::Metamodel::DataTypes::DataType |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageBuildingBlock |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |

| | |
|---|---|
| Ordered | false |

**Constraints**

— MessageBuildingBlockHasExactlyOneType

**Description**

A MessageBuildingBlock shall have exactly one of each of the following: simpleType or complexType.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageBuildingBlock |
| Language | OCL2.0 |
| Body | complexType->size() + simpleType->size() = 1 |

### B.2.8.8 Metaclass MessageChoreography

**Description**

Precise and complete description of a MessageInstance exchange within a BusinessTransaction, describing the sequence and correlation of MessageInstances within a conversation, including the constraints on the interaction between Participants.

NOTE Every BusinessTransaction contains its own MessageChoreography.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

**Properties**

— businessTransactionTrace

**Description**

The BusinessTransactionTrace from the MessageChoreography to the BusinessTransaction.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessTransactionTrace |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageChoreography |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— messageDefinition

**Description**

The MessageDefinition that is used in a MessageChoreography.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageChoreography |
| Default Value | |
| Multiplicity | 1..* |
| Aggregation | shared |
| Ordered | false |

**Constraints**

— UniqueName

**Description**

All MessageChoreographies shall have a unique name within the Catalogue.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageChoreography |
| Language | OCL2.0 |
| Body | topLevelCatalogueEntry -> select(oclIsKindOf(SyntaxMessageScheme)) -> isUnique(name) |

### B.2.8.9 Metaclass MessageComponent

**Description**

Re-usable Dictionary Item, composed of a sequence of MessageElements, that is a building block for assembling MessageDefinitions.

EXAMPLE Trade Details (which contains a number of the properties of the related BusinessComponent “Trade”).

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::LogicalLevel::MessageComponentType |

### B.2.8.10 Metaclass MessageComponentType

**Description**

MessageComponent, ExternalSchema or ChoiceComponent.

NOTE 1 When a MessageComponentType has a business meaning it is linked to a BusinessComponent.

NOTE 2 MessageComponentTypes are a category of MessageConcepts, stored in the DataDictionary.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel::LogicalLevel |

| Applied Stereotype | metaclass |
|---|---|
| Superclass | ISO20022::Metamodel::MessageConcept<br>ISO20022::Metamodel::TopLevelDictionaryEntry |

**Properties**

— isTechnical

**Description**

Property of a MessageComponentType indicating the absence of a semantic relationship to a BusinessComponent, i.e. it is not derived from a BusinessComponent.

NOTE 3 A MessageComponentType is considered to be 'technical' if it has been created for 'message specific' reasons.

EXAMPLE Extension MessageComponents that allow any MessageComponent to be extended from the MessageDefinition.

| Type | ISO20022::TypeLibrary::XMLSchema::boolean |
|---|---|
| Owner | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Default Value | trace->isEmpty() |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageBuildingBlock

**Description**

The MessageBuildingBlock which is the context for this MessageComponentType.

| Type | ISO20022::Metamodel::LogicalLevel::MessageBuildingBlock |
|---|---|
| Owner | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— messageElement

**Description**

A semantic property of a MessageComponentType.

| Type | ISO20022::Metamodel::LogicalLevel::MessageElement |
|---|---|
| Owner | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Default Value | |

| | |
|---|---|
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | true |

— trace

**Description**

The derivation of a MessageComponentType from a BusinessComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessComponentTrace |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— MessageElementsHaveUniqueNames

**Description**

All MessageElements contained in this MessageComponentType shall have different names.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Language | OCL2.0 |
| Body | messageElement->forAll(el1,el2 \| el1 <> el2 implies el1.name <> el2.name) |

— technicalElement

**Description**

All the elements belonging to a technical MessageComponentType shall be technical.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Language | OCL2.0 |
| Body | self.isTechnical implies element-> forAll(messageElement\|messageElement.isTechnical) |

— UniqueName

**Description**

All MessageComponentTypes shall have a unique name within the Dictionary.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Language | OCL2.0 |
| Body | topLevelDictionaryEntry -> select(oclIsKindOf(SyntaxMessageScheme)) -> isUnique(name) |

#### B.2.8.11 Metaclass MessageDefinition

**Description**

Formal description of the structure of a MessageInstance.

NOTE 1 The MessageDefinition is built as a tree structure of MessageComponentTypes and DataTypes. A MessageDefinition is uniquely identified in the BusinessProcessCatalogue.

NOTE 2 A MessageDefinition can have several market practices

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— businessArea

**Description**

The BusinessArea to which this MessageDefinition belongs.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::BusinessArea |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— choreography

**Description**

The MessageChoreography to which the MessageDefinition belongs.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageChoreography |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Default Value | |
| Multiplicity | 1..* |
| Aggregation | none |

| | |
|---|---|
| Ordered | false |

— derivation

**Description**

All of the MessageDefinitionTraces that derive SyntaxMessageSchemes from one MessageDefinition.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalToPhysicalTransformation::MessageDefinitionTrace |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— messageBuildingBlock

**Description**

A MessageBuildingBlock belonging to this MessageDefinition.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageBuildingBlock |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Default Value | |
| Multiplicity | 1..* |
| Aggregation | composite |
| Ordered | true |

— messageDefinitionIdentifier

**Description**

The MessageDefinitionIdentifier for this MessageDefinition.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageDefinitionIdentifier |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | composite |
| Ordered | false |

— messageSet

**Description**

The MessageSet to which the MessageDefinition belongs.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageSet |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— rootElement

**Description**

A property of MessageDefinition that specifies the top level element of the MessageDefinition.

NOTE 3 In ISO 20022 XML Schemas, this results in the only globally declared element.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::Name |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— trace

**Description**

All of the MessageTypeTraces from one MessageDefinition that are traced to different MessageTransmissions.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::MessageTypeTrace |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

**Constraints**

— BusinessAreaNameMatch

**Description**

The businessArea of the MessageDefinitionIdentifier of this MessageDefinition is equal to the code of the BusinessArea that contains this MessageDefinition.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Language | OCL2.0 |
| Body | businessArea.code = messageDefinitionIdentifier.businessArea |

### B.2.8.12 Metaclass MessageDefinitionIdentifier

**Description**

Unique identification of a MessageDefinition within the ISO 20022 namespace, identifying the BusinessArea to which the MessageDefinition belongs, the Message Functionality it covers, its flavour and its version.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— businessArea

**Description**

Part of a MessageDefinitionIdentifier describing the BusinessArea to which the MessageDefinition using this MessageDefinitionIdentifier belongs.

NOTE 1 This information is derived from the BusinessArea to which this MessageDefinition is associated.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::NMTOKEN |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinitionIdentifier |
| Default Value | messageDefinition.businessArea.code |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— flavour

**Description**

A variation of a messageFunctionality, whereby MessageDefinitions that are based on this MessageDefinition are compliant with the MessageDefinition from which this MessageDefinition is derived.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::token |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinitionIdentifier |

| | |
|---|---|
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— messageFunctionality

**Description**

Function and purpose for which a MessageInstance described by a MessageDefinition can be used.

NOTE 2 MessageDefinitions in existing industry MessageSets are often multi-functional, meaning that they can be used for multiple purposes. The ISO 20022 Method contains guidelines to avoid such multi-functionality.

NOTE 3 An example of multi-functionality is the ISO 15022-1:1999 message “MT 502”, which can be used as an order to buy securities, as an order to sell securities, to cancel a previously placed order, or to change a previously placed order.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::token |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinitionIdentifier |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— version

**Description**

Identifies the version of the MessageDefinition.

NOTE 4 Versions start with '01' whereby a new version = old version + 1.

NOTE 5 A new version implies that the old version will become deprecated. This distinguishes the concept of flavour where a new flavour does not imply/impose the deprecation of another MessageDefinition.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::token |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageDefinitionIdentifier |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

### B.2.8.13 Metaclass MessageElement

**Description**

Characteristic of a MessageComponent/ChoiceComponent, which has a unique meaning within the scope of that MessageComponent/ChoiceComponent.

EXAMPLE 1 Trade Date and Time (as part of the MessageComponent “Trade Details”).

NOTE 1 MessageElements are a category of MessageConcepts. They are stored in the DataDictionary where they are owned by a particular MessageComponent/ChoiceComponent. Their meaning can only be described unambiguously in combination with that MessageComponent/ChoiceComponent.

| | |
|---|---|
| Abstract | true |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::MessageConcept ISO20022::Metamodel::Repository Concept |

**Properties**

— businessComponentTrace

**Description**

The trace of a MessageElement to the BusinessComponent from which it is derived.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessComponentTrace |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— businessElementTrace

**Description**

The trace of a MessageElement to the BusinessElement from which the MessageElement is derived.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualToLogicalTransformation::BusinessElementTrace |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

— cardinality

**Description**

Property of a MesasageElement describing the number of allowed occurrences it may have in a particular MessageComponent or ChoiceComponent.

| | |
|---|---|
| Type | ISO20022::Metamodel::Cardinality |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— componentContext

**Description**

The MessageComponentType that describes the context within which the MessageElement is defined.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageComponentType |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— isDerived

**Description**

Property of a MessageElement that specifies whether a MessageElement can be computed using other MessageElements. It is shown for clarity even though it adds no semantic information.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::boolean |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Default Value | false |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— isTechnical

**Description**

Property of a MessageElement indicating whether it has a semantic relationship to a BusinessComponent/BusinessElement, i.e. whether it is derived from a BusinessComponent/BusinessElement.

NOTE 2 A MessageElement is considered to be 'technical' if it has been created for 'message specific' reasons.

EXAMPLE 2 Page number, certain message references.

| | |
|---|---|
| Type | ISO20022::TypeLibrary::XMLSchema::boolean |

| | |
|---|---|
| Owner | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Default Value | trace->isEmpty() |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— NoMoreThanOneTrace

**Description**

A MessageElement may not trace to both a BusinessComponent and a BusinessElement.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageElement |
| Language | OCL2.0 |
| Body | businessComponentTrace->size( ) + businessElementTrace->size( ) <= 1 |

#### B.2.8.14 Metaclass MessageSet

**Description**

Set of MessageDefinitions

NOTE MessageDefinitions within a MessageSet do not have to belong to the same BusinessArea.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

**Properties**

— generatedSyntax

**Description**

Identification of the syntax for a specific EncodingScheme.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::Syntax |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageSet |
| Default Value | |
| Multiplicity | 1..* |
| Aggregation | none |
| Ordered | false |

— messageDefinition

**Description**

The MessageDefinition that belongs to the MessageSet.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageSet |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | shared |
| Ordered | false |

— validEncoding

**Description**

The set of encodings considered ISO 20022 valid for this MessageSet.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::Encoding |
| Owner | ISO20022::Metamodel::LogicalLevel::MessageSet |
| Default Value | |
| Multiplicity | 1..* |
| Aggregation | none |
| Ordered | false |

**Constraints**

— GeneratedSyntaxDerivation

**Description**

The generated syntax is derived from the syntax for a validEncoding.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageSet |
| Language | OCL2.0 |
| Body | generatedSyntax=validEncoding.syntax |

— UniqueName

**Description**

All MessageSets must have a unique name within the Catalogue.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::MessageSet |
| Language | OCL2.0 |

| | |
|---|---|
| Body | topLevelCatalogueEntry -> select(oclIsKindOf(SyntaxMessageScheme)) -> isUnique(name) |

#### B.2.8.15 Metaclass Syntax

**Description**

A particular syntax for a MessageInstance.

EXAMPLE 1 W3C XML Schema 1.0

EXAMPLE 2 ISO ASN.1

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— generatedFor

**Description**

The scheme in which a syntax is encoded.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageSet |
| Owner | ISO20022::Metamodel::LogicalLevel::Syntax |
| Default Value | |
| Multiplicity | * |
| Aggregation | none |
| Ordered | false |

— possibleEncodings

**Description**

The set of possible encodings for a given Syntax.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::Encoding |
| Owner | ISO20022::Metamodel::LogicalLevel::Syntax |
| Default Value | |
| Multiplicity | 1..* |
| Aggregation | none |
| Ordered | false |

**Constraints**

— GeneratedForDerivation

**Description**

The syntax that is valid for a MessageSet, based on the Encoding used.

| | |
|---|---|
| Context | ISO20022::Metamodel::LogicalLevel::Syntax |
| Language | OCL2.0 |
| Body | generatedFor=possibleEncodings.messageSet |

## B.2.9 Package ISO20022::Metamodel::LogicalLevel::Reversing

### B.2.9.1 Metaclass ConvergenceDocumentation

**Description**

Documentation set showing relations between ISO 20022 MessageDefinitions, MessageComponentTypes, MessageElements, BusinessComponents and/or BusinessElements and items defined in other IndustryMessageSets.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel::Reversing |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

### B.2.9.2 Metaclass IndustryMessageSet

**Description**

Set of non-ISO 20022 compliant messages that is defined and used by part of the financial industry.
EXAMPLE The set of FIX v5 messages.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel::Reversing |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

### B.2.9.3 Metaclass ISO15022MessageSet

**Description**

IndustryMessageSet constructed according to the rules defined in ISO 15022-1 and ISO 15022-2, which is stored in the ISO 15022 Catalogue of Messages.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalLevel::Reversing |
| Applied Stereotype | metaclass |

| | |
|---|---|
| Superclass | ISO20022::Metamodel::LogicalLevel::Reversing::IndustryMessageSet |

## B.2.10 Package ISO20022::Metamodel::LogicalToPhysicalTransformation

### B.2.10.1 Metaclass MessageDefinitionTrace

**Description**

Relationship between a MessageDefinition and its physical implementation as a SyntaxMessageScheme.

NOTE This relationship is explained in ISO 20022-4.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::LogicalToPhysicalTransformation |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::Trace |

**Properties**

— source

**Description**

The MessageDefinition to which the SyntaxMessageScheme is traced.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalLevel::MessageDefinition |
| Owner | ISO20022::Metamodel::LogicalToPhysicalTransformation::MessageDefinitionTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— syntaxMessageScheme

**Description**

The SyntaxMessageScheme that is traced to the MessageDefinition.

| | |
|---|---|
| Type | ISO20022::Metamodel::PhysicalLevel::SyntaxMessageScheme |
| Owner | ISO20022::Metamodel::LogicalToPhysicalTransformation::MessageDefinitionTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

## B.2.11 Package ISO20022::Metamodel::PhysicalLevel

### B.2.11.1 Metaclass MessageInstance

**Description**

Instance of MessageDefinition, containing a set of structured information exchanged between BusinessRoles, in the scope of a BusinessTransaction.

EXAMPLE Notice Of Execution, Order To Buy, Credit Transfer.

NOTE A MessageInstance is expected to be valid against the related MessageDefinition in the ISO 20022 Repository. This implies validity against the SyntaxMessageScheme as well as validity against the Constraints and market practices that are registered for this MessageDefinition.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::PhysicalLevel |
| Applied Stereotype | metaclass |
| Superclass | |

**Properties**

— specification

**Description**

The SyntaxScheme instantiated by this MessageInstance.

| | |
|---|---|
| Type | ISO20022::Metamodel::PhysicalLevel::SyntaxMessageScheme |
| Owner | ISO20022::Metamodel::PhysicalLevel::MessageInstance |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— transportMessage

**Description**

The TransportMessage that contains the MessageInstance.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::MessageTransport::TransportMessage |
| Owner | ISO20022::Metamodel::PhysicalLevel::MessageInstance |
| Default Value | |
| Multiplicity | 1..* |
| Aggregation | none |
| Ordered | false |

### B.2.11.2 Metaclass SyntaxMessageScheme

**Description**

Syntax processable notation used to define the structure of a MessageInstance in a particular syntax.

NOTE 1 In case of XML, the representation may, for instance, be an XML DTD or an XML Schema and can then be used as a validation tool for MessageInstances.

NOTE 2 Syntax message representations are stored in the BusinessProcessCatalogue.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::PhysicalLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

**Properties**

— messageDefinitionTrace

**Description**

The trace from the SyntaxMessageScheme to the MessageDefinition.

| | |
|---|---|
| Type | ISO20022::Metamodel::LogicalToPhysicalTransformation::MessageDefinitionTrace |
| Owner | ISO20022::Metamodel::PhysicalLevel::SyntaxMessageScheme |
| Default Value | |
| Multiplicity | 0..1 |
| Aggregation | none |
| Ordered | false |

**Constraints**

— UniqueName

**Description**

All SyntaxMessageSchemes shall have a unique name within the Catalogue.

| | |
|---|---|
| Context | ISO20022::Metamodel::PhysicalLevel::SyntaxMessageScheme |
| Language | OCL2.0 |
| Body | topLevelCatalogueEntry -> select(oclIsKindOf(SyntaxMessageScheme)) -> isUnique(name) |

## B.2.12 Package ISO20022::Metamodel::ScopeLevel

### B.2.12.1 Metaclass BusinessProcess

**Description**

Unrealized definition of the business activities undertaken by BusinessRoles within a BusinessArea whereby each BusinessProcess fulfils one type of business activity and whereby a BusinessProcess might include and extend other BusinessProcesses.

EXAMPLE Securities ordering, trade matching.

NOTE 1 A BusinessProcess can contain other BusinessProcesses such as in a hierarchical structure.

NOTE 2 BusinessProcesses are stored in the BusinessProcessCatalogue.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ScopeLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::TopLevelCatalogueEntry |

**Properties**

— businessProcessTrace

**Description**

The BusinessProcessTrace that is tracing to the BusinessProcess.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeToConceptualTransformation::BusinessProcessTrace |
| Owner | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— businessRole

**Description**

The BusinessRole that plays a role in BusinessProcess that owns it.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeLevel::BusinessRole |
| Owner | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | composite |
| Ordered | false |

— extended

**Description**

Specifies the BusinessProcess that is extended by another BusinessProcess.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Owner | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— extender

**Description**

Specifies the BusinessProcess that is extending another BusinessProcess.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Owner | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— included

**Description**

Specifies the BusinessProcess that is included by another BusinessProcess.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Owner | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

— includer

**Description**

Specifies the BusinessProcess that is including another BusinessProcess.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Owner | ISO20022::Metamodel::ScopeLevel::BusinessProcess |

| | |
|---|---|
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

**Constraints**

— UniqueName

**Description**

All BusinessProcesses shall have a unique name within the Catalogue.

| | |
|---|---|
| Context | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Language | OCL2.0 |
| Body | topLevelCatalogueEntry -> select(oclIsKindOf(SyntaxMessageScheme)) -> isUnique(name) |

### B.2.12.2 Metaclass BusinessRole

**Description**

Functional role played by a business actor in a particular BusinessProcess or BusinessTransaction.
EXAMPLE Account owner, buyer.

NOTE 1 BusinessRoles are a category of BusinessConcepts and are stored in the DataDictionary.
NOTE 2 A business actor can play different BusinessRoles in different BusinessProcesses.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ScopeLevel |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::RepositoryConcept |

**Properties**

— businessProcess

**Description**

The BusinessProcess that owns the BusinessRole.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Owner | ISO20022::Metamodel::ScopeLevel::BusinessRole |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— businessRoleTrace

**Description**

The BusinessRoleTrace for a BusinessRole.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeToConceptualTransformation::BusinessRoleTrace |
| Owner | ISO20022::Metamodel::ScopeLevel::BusinessRole |
| Default Value | |
| Multiplicity | 0..* |
| Aggregation | none |
| Ordered | false |

## B.2.13 Package ISO20022::Metamodel::ScopeToConceptualTransformation

### B.2.13.1 Metaclass BusinessProcessTrace

**Description**

Relationship between a BusinessTransaction and the BusinessProcess on which this BusinessTransaction is based.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ScopeToConceptualTransformation |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::Trace |

**Properties**

— businessProcess

**Description**

The BusinessProcess to which the BusinessProcessTrace traces.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeLevel::BusinessProcess |
| Owner | ISO20022::Metamodel::ScopeToConceptualTransformation::BusinessProcessTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— businessTransaction

**Description**

The BusinessTransaction that is traced using a BusinessProcessTrace.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::BusinessTransaction |
| Owner | ISO20022::Metamodel::ScopeToConceptualTransformation::BusinessProcessTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

#### B.2.13.2 Metaclass BusinessRoleTrace

**Description**

Relationship between a Participant in a BusinessTransaction and a BusinessRole identified in the BusinessProcess from which the BusinessTransaction is derived.

| | |
|---|---|
| Abstract | false |
| Owner | ISO20022::Metamodel::ScopeToConceptualTransformation |
| Applied Stereotype | metaclass |
| Superclass | ISO20022::Metamodel::Trace |

**Properties**

— businessRole

**Description**

The BusinessRole to which the BusinessRoleTrace traces.

| | |
|---|---|
| Type | ISO20022::Metamodel::ScopeLevel::BusinessRole |
| Owner | ISO20022::Metamodel::ScopeToConceptualTransformation::BusinessRoleTrace |
| Default Value | |
| Multiplicity | 1 |
| Aggregation | none |
| Ordered | false |

— participant

**Description**

The Participant that is traced using a BusinessRoleTrace.

| | |
|---|---|
| Type | ISO20022::Metamodel::ConceptualLevel::Dynamic::Participant |
| Owner | ISO20022::Metamodel::ScopeToConceptualTransformation::BusinessRoleTrace |
| Default Value | |
| Multiplicity | 1 |

| | |
|---|---|
| Aggregation | none |
| Ordered | false |

## B.3 Metamodel XMI

The metamodel in XMI is posted on http://www.iso20022.org/

# Bibliography

[1] ISO/TR 7775, *Securities — Scheme for message types*

[2] ISO 8601, *Data elements and interchange formats — Information interchange — Representation of dates and times*

[3] ISO 9362, *Banking — Banking telecommunication messages — Business identifier code (BIC)*

[4] ISO 11179-1, *Information technology — Metadata registries (MDR) — Part 1: Framework*

[5] ISO 15022 (all parts), *Securities — Scheme for messages (Data Field Dictionary)*

[6] ISO/IEC 19502:2005, *Information Technology— Meta Object Facility (MOF)*

[7] UML (Unified Modeling Language), Version 2 — Object Management Group

[8] MOF (Meta Object Facility) Version 2.0 — Object Management Group

[9] XML (Extensible Markup Language) 1.0 (Second Edition) W3C Recommendation 6 October 2000 — World Wide Web Consortium

[10] W3C Recommendation: XML Schema Part 1: Structures Second Edition (28 October 2004)

**ICS 03.060**
Price based on 152 pages

*This page deliberately left blank*

# British Standards Institution (BSI)

BSI is the national body responsible for preparing British Standards and other standards-related publications, information and services.

BSI is incorporated by Royal Charter. British Standards and other standardization products are published by BSI Standards Limited.

## About us

We bring together business, industry, government, consumers, innovators and others to shape their combined experience and expertise into standards -based solutions.

The knowledge embodied in our standards has been carefully assembled in a dependable format and refined through our open consultation process. Organizations of all sizes and across all sectors choose standards to help them achieve their goals.

## Information on standards

We can provide you with the knowledge that your organization needs to succeed. Find out more about British Standards by visiting our website at bsigroup.com/standards or contacting our Customer Services team or Knowledge Centre.

## Buying standards

You can buy and download PDF versions of BSI publications, including British and adopted European and international standards, through our website at bsigroup.com/shop, where hard copies can also be purchased.

If you need international and foreign standards from other Standards Development Organizations, hard copies can be ordered from our Customer Services team.

## Subscriptions

Our range of subscription services are designed to make using standards easier for you. For further information on our subscription products go to bsigroup.com/subscriptions.

With **British Standards Online (BSOL)** you'll have instant access to over 55,000 British and adopted European and international standards from your desktop. It's available 24/7 and is refreshed daily so you'll always be up to date.

You can keep in touch with standards developments and receive substantial discounts on the purchase price of standards, both in single copy and subscription format, by becoming a **BSI Subscribing Member**.

**PLUS** is an updating service exclusive to BSI Subscribing Members. You will automatically receive the latest hard copy of your standards when they're revised or replaced.

To find out more about becoming a BSI Subscribing Member and the benefits of membership, please visit bsigroup.com/shop.

With a **Multi-User Network Licence (MUNL)** you are able to host standards publications on your intranet. Licences can cover as few or as many users as you wish. With updates supplied as soon as they're available, you can be sure your documentation is current. For further information, email bsmusales@bsigroup.com.

**BSI Group Headquarters**

389 Chiswick High Road London W4 4AL UK

## Revisions

Our British Standards and other publications are updated by amendment or revision.

We continually improve the quality of our products and services to benefit your business. If you find an inaccuracy or ambiguity within a British Standard or other BSI publication please inform the Knowledge Centre.

## Copyright



## Useful Contacts:

**Customer Services**
**Tel:** +44 845 086 9001
**Email (orders):** orders@bsigroup.com
**Email (enquiries):** cservices@bsigroup.com

**Subscriptions**
**Tel:** +44 845 086 9001
**Email:** subscriptions@bsigroup.com

**Knowledge Centre**
**Tel:** +44 20 8996 7004
**Email:** knowledgecentre@bsigroup.com

**Copyright & Licensing**
**Tel:** +44 20 8996 7070
**Email:** copyright@bsigroup.com

bsi.

...making excellence a habit.™