



DPC: 25/30461189 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000

Fax: +44 (0)20 8996 7400

www.bsigroup.com

Date: 03 March 2025

Origin: International

Latest date for receipt of comments: 26 April 2025

Project No. 2022/01849

Responsible committee: IST/12 Financial services

Interested committees:

Title: Draft BS ISO 20022-1 Financial services - Universal financial industry message scheme. Part 1: Metamodel.

Please notify admin.start@bsigroup.com if you are aware of any keywords that might assist in clarifying or identifying the standard or of the content of this standard:

- i) has any issues related to 3rd party IPR, patent or copyright
- ii) affects other national standard(s)
- iii) requires additional national guidance or information

WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD. THIS DRAFT IS NOT CURRENT BEYOND 26 April 2025

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop <http://shop.bsigroup.com> or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from <https://standardsdevelopment.bsigroup.com/>.

E-mail: admin.start@bsigroup.com

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at <https://standardsdevelopment.bsigroup.com/>. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx
------------------	------------------------

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex (e.g. 3.1)	Paragraph/ Figure/Table/Note	Type of comment	Comment (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	



DRAFT International Standard

ISO/DIS 20022-1

Financial services — Universal financial industry message scheme —

Part 1: Metamodel

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

Partie 1: Métamodèle

ICS: 03.060

ISO/TC 68/SC 9

Secretariat: **AFNOR**

Voting begins on:
2025-03-03

Voting terminates on:
2025-05-26

This document is circulated as received from the committee secretariat.

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

Reference number
ISO/DIS 20022-1:2025(en)

Full document details: 25/30461189 DC

© ISO 2025



Documents

Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Type Library	11
5 Metamodel packages	14
5.1 General	14
5.2 The metamodel's use of ISO20022::TypeLibrary	14
5.3 Levels	15
5.3.1 Scope level	16
5.3.2 Conceptual level	17
5.3.3 Logical level	19
5.3.4 Physical level	21
5.4 Models Transformation	22
5.4.1 Scope to Conceptual	23
5.4.2 Conceptual to Logical	23
5.4.3 Logical to Physical	24
6 Repository	25
6.1 ISO 20022 Repository structure	25
6.1.1 Overview	25
6.2 DataDictionary	27
6.2.1 Overview	27
6.2.2 List of Dictionary Items	28
6.2.3 DictionaryItem Registration Status	31
6.2.4 DictionaryItems description information	32
6.2.5 DataDictionary life cycle	32
6.3 BusinessProcessCatalogue	32
6.3.1 Overview	32
6.3.2 List of BusinessProcessCatalogueItems	33
6.3.3 CatalogueItem Registration Status	33
6.3.4 CatalogueItem description information	33
6.3.5 BusinessProcessCatalogue life cycle	34
7 Registration	34
8 Repository Access	34
8.1 General	34
8.2 Repository output types	34
8.3 Output format	34
Annex A (normative) Type library	35
Annex B (normative) Metamodel	49
Bibliography	165

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 third edition cancels and replaces the second edition (ISO 20022-1:2013), which has been 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: Conceptual interoperability and reverse engineering*
- *Part 6: Message transport characteristics*
- *Part 7: Registration*
- *Part 8: ASN.1 generation*
- *Part 9: Syntax Generation Requirements and Rules*

ISO 20022-1, ISO 20022-2, ISO 20022-3, ISO 20022-4, ISO 20022-5, ISO 20022-6, ISO 20022-7, ISO 20022-8 and ISO 20022-9 will be implemented by the Registration Authority, at which time support for the concepts set out within them will be effective. The implementation plan should be agreed between the Registration Authority, Registration Management Group and Technical Support Group. Users and potential users of the ISO 20022 series are encouraged to familiarize themselves with the 2025 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.

The main technical changes in this edition are:

- **Changes**
 - Use of ISO/IEC 11404:2007 General Purpose Data Types instead of XML schema, except XML Schema temporal types remain.
 - Code Set trace to Business Code Set, formalising the two level approach used in practice.

- Top level entries across the catalogue and dictionary have unique names.
- Clarifications
 - Cardinality specified and updated related constraints.
 - Lengths of Binary, String and Text types are positiveIntegers.
 - MessageAssociationEnd aggregation to align with BusinessAssociationEnd.
 - Choreography associated indirectly with MessageDefinitions via MessageSet, instead of directly.
- Additions
 - Defined term Conformance to replace use of "compliance"
 - SyntaxMessageScheme linked to Syntax.
 - Import relationship amongst BusinessTransactions.
 - Conversation trace to Choreography, as a conversation is an instance of a choreography.
 - Conversation aggregates MessageInstances to MessagingEndpoints, as per its definition.
 - Pointer data type to enable direct references within each message instance.
 - Binary types specify preferred text representation.
 - Minor version property added to MessageDefinitionIdentifier
 - Path of traces from BusinessComponent to BusinessElement.
 -
 - Interface having Property and Operation that use MessageDefinition.
 - Protocol having Port that provide or require Interface. Port traces to Participant. Protocol links to Choreography.
 - Kind and Type of Constraint.
 - Version relationships amongst RepositoryConcept.

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 model artefacts are stored in a central repository. The repository and physical level artefacts are exposed in a publicly accessible location, such as a website, serviced by a Registration Authority.

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 MessageDefinitions and BusinessProcesses, 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 business concept model interoperability, and logical model alignment and reverse engineering.
- ISO 20022-6 covers message transport characteristics that define the quality of service required by the BusinessProcess 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.
- ISO 20022-9 describes generic syntax generation guidelines which are used to define specific syntax generation rules, such as JSON schema, etc.

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 MessageSets Conforming 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 — Part 2: UML profile*

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

ISO/IEC 11404:2007, *Information technology — General-Purpose Datatypes (GPD)*

754-2019 - IEEE Standard for Floating-Point Arithmetic

ISO 639:2023, *Code for individual languages and language groups*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply. Defined terms of this standard are in PascalCase and will use PascalCase throughout the standard.

3.1

Address

identification and efficient resolution to the location of a MessagingEndpoint

Note 1 to entry: 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

API (Application Programming Interface)

a set of well-defined methods, functions, protocols, routines or commands which application software uses with facilities of programming languages to invoke services

Note 1 to entry: An API is available for different types of software, including Web-based system/ecosystem.

[SOURCE: ISO TS 23029:2020 <https://www.iso.org/obp/ui/en/#iso:std:iso:ts:23029:ed-1:v1:en>]

3.4

ApiCall

representation of a single transaction or interaction between systems allowing a client to request data or perform an action on a server

3.5

ApiEndpoint

logical location where an API receives requests about a specific API Resource

3.6

ApiMethod

way to manage an API Resource

3.7

binary

any set of values drawn from the value space of 'octetstring', the datatype of variable-length encodings using 8-bit codes, as specified by ISO/IEC 11404:2007 General Purpose Datatypes

3.8

Boolean

any set of values drawn from the value space of 'Boolean', two-value logic as specified by ISO/IEC 11404:2007 General Purpose Datatypes

3.9

BroadcastList

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

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

Note 2 to entry: "Set" means the list of Addresses is unordered and each Address may only be present once.

3.10

BusinessActivity

decisions or actions by Participants which may initiate the sending of a MessageTransmission or be triggered by receiving a MessageTransmission

Note 1 to entry: A BusinessActivity may also result in exchange of information at an API endpoint.

3.11

BusinessArea

set of strongly related BusinessProcesses that provide a self-standing business value to a set of BusinessRoles

EXAMPLE Securities pre-trade, payment initiation.

Note 1 to entry: BusinessAreas are stored in the BusinessProcessCatalogue.

3.12

BusinessAssociation

relation between two BusinessComponents

EXAMPLE The servicing of an account by a party.

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

Note 2 to entry: There can be several BusinessAssociations between two particular BusinessComponents.

3.13

BusinessAssociationEnd

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

3.14

BusinessAttribute

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

EXAMPLE AccountIdentification, PhoneNumber.

3.15

BusinessCodeSet

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

3.16

BusinessComponent

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

EXAMPLE Account, trade, party.

Note 1 to entry: BusinessComponents are a category of BusinessConcepts. They are stored in the DataDictionary.

Note 2 to entry: A BusinessComponent can have one or more semantic relations with other BusinessComponents.

3.17

BusinessComponentTrace

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

3.18

BusinessConcept

a DataDictionary item of meaning to the business expressed as a BusinessComponent, BusinessElement or BusinessAssociation

3.19

business domain

broad area of an industry with its own regulatory framework(s), usually supported by specialist areas within organisations with its own terminology

EXAMPLE Card payments and related transactions, payments and cash management, trade services, securities, foreign exchange, bank loan and deposits, derivatives, commodities.

3.20

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.

3.21

BusinessElementTrace

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

3.22

BusinessProcess

name and definition of a set of related business activities undertaken by BusinessRoles within a BusinessArea which fulfils a business objective

EXAMPLE Securities ordering, trade matching.

Note 1 to entry: A BusinessProcess can contain other BusinessProcesses such as in a hierarchical structure.

Note 2 to entry: BusinessProcesses are stored in the BusinessProcessCatalogue.

Note 3 to entry: The definition of a scope level BusinessProcess may include or extend other BusinessProcesses.

3.23

BusinessProcessCatalogue

that part of the ISO 20022 Repository which contains all items related to BusinessProcess and BusinessTransaction

Note 1 to entry: It contains related items from the BusinessArea down to the MessageDefinitions and their physical implementation.

Note 2 to entry: CatalogueItem is used as a synonym for entries in the BusinessProcessCatalogue.

3.24

BusinessProcessTrace

relationship between a BusinessTransaction and the BusinessProcess on which the BusinessTransaction is based

3.25

BusinessRole

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

EXAMPLE Account owner, buyer.

Note 1 to entry: BusinessRoles are a category of BusinessConcepts and are stored in the DataDictionary.

Note 2 to entry: A business actor can play different BusinessRoles in different BusinessProcesses.

3.26

BusinessRoleTrace

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

3.27

BusinessTransaction

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

Note 1 to entry: It is typically based on the use of MessageInstances.

3.28

BusinessTransactionTrace

relationship between a BusinessTransaction and its physical implementation

3.29

ChoiceComponent

re-usable DictionaryItem that is a building block for assembling MessageDefinitions, composed of a choice of MessageElements

Note 1 to entry: ChoiceComponents are stored in the DataDictionary.

3.30

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

3.31

CodeSet

a type that has a finite set of valid Code values, as specified by the State datatype of ISO/IEC 11404:2007 General-Purpose Datatypes

3.32

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

3.33

Conformance

adherence to requirements, rules, guidelines and advice

Note 1 to entry: "Conformant", "Conforming", "Conform" forms of the term carries the same definition

EXAMPLE 1 A model design should be Conformant with ISO 20022-1 and ISO 20022-3

EXAMPLE 2 Schema generation is Conformant with its specification

3.34

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.

3.35

ConvergenceDocumentation

documentation set showing relations between ISO 20022 MessageDefinitions, MessageComponentTypes, MessageElements, BusinessComponents and/or BusinessElements and items defined in other IndustryMessageSets

3.36

Conversation

exchange of one or more MessageInstances amongst MessagingEndpoints

Note 1 to entry: 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.

3.37

DataDictionary

part of the ISO 20022 Repository that contains all items that can be re-used during BusinessProcess modelling and message definition activities

Note 1 to entry: The DataDictionary therefore contains BusinessConcepts, MessageConcepts and DataTypes.

3.38

Data Type

representation of a set of values without identity

3.39

date

any set of values drawn from the value space of 'time' with a time-unit of 'day', as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.40

dateTime

any set of values drawn from the value space of 'time', as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.41**day**

any set of values drawn from the value space of 'dayof-Month' operation of time datatype as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.42**decimal**

any set of values drawn from the value space of 'scaled' with a radix of '10', as specified as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.43**DictionaryItem**

BusinessConcept, DataType, MessageConcept items

3.44**duration**

any set of values drawn from the value space of 'time interval', as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.45**Encoding**

wire format of message instances

Note 1 to entry: An encoding specifies the format of message instances that are structured according to a Syntax Message Scheme.

EXAMPLE 1 ASN.1 PER

EXAMPLE 2 ASN.1 XER

EXAMPLE 3 XSD XML

EXAMPLE 4 JSON (IETF RFC 8259)

EXAMPLE 5 RDF TTL

3.46**ExternalSchema**

reusable DictionaryItem 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.

3.47**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

3.48**indicator**

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

EXAMPLE 1 PlusOrMinusIndicator can be plus or minus.

EXAMPLE 2 DebitCreditIndicator can be debit or credit.

3.49**IndustryMessageSet**

set of non-ISO 20022 Conformant messages, which is defined and used by part of the financial industry

EXAMPLE The set of FIX messages.

3.50

Interface

specification of Operations and Properties used for communication

3.51

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

3.52

MessageAssociationEnd

type of MessageElement that specifies the role of a MessageAssociation

3.53

MessageAttribute

type of MessageElement which is either a DataType or a MessageComponentType

3.54

MessageBuildingBlock

characteristic of a MessageDefinition that has a unique meaning within the scope of that MessageDefinition

Note 1 to entry: MessageBuildingBlocks are not reused, since they only have meaning within a specific MessageDefinition.

3.55

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 1 to entry: Every BusinessTransaction contains its own MessageChoreography.

Note 2 to entry: Includes MessageInstances that flow to/from API interfaces.

3.56

MessageComponent

re-usable DictionaryItem 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".

3.57

MessageComponentType

MessageComponent, ExternalSchema or ChoiceComponent

Note 1 to entry: When a MessageComponentType has a business meaning it is linked to a BusinessComponent.

Note 2 to entry: MessageComponentTypes are a category of MessageConcepts and are stored in the DataDictionary.

3.58

MessageConcept

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

3.59

MessageDefinition

formal description of the structure of a MessageInstance

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

Note 2 to entry: A MessageDefinition can have several market practices.

3.60

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

3.61

MessageDefinitionTrace

relationship between a MessageDefinition and its physical implementation as a SyntaxMessageScheme

Note 1 to entry: This relationship is explained in ISO 20022-4.

3.62

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 1 to entry: 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.

3.63

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 1 to entry: 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.

Note 2 to entry: A MessageInstance includes an exchange of information from one participant to another.

3.64

MessageSet

set of MessageDefinitions

Note 1 to entry: MessageDefinitions within a MessageSet do not have to belong to the same BusinessArea.

3.65

MessageTransmission

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

3.66

MessageTransportMode

group of settings for the values for the MessageTransportCharacteristics properties

Note 1 to entry: A MessageTransportMode is named and registered in the ISO 20022 Repository. Each MessageTransportCharacteristic is given a value.

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

3.67

MessageTransportSystem

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

Note 1 to entry: The MessageTransportSystem is responsible for delivering Transport Messages to each Addressee.

Note 2 to entry: 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. MessageTransportSystems can be chained together into a single MessageTransportSystem

3.68

MessageTypeTrace

relationship between a MessageTransmission in a BusinessTransaction and its corresponding MessageDefinition

3.69

MessagingEndpoint

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

Note 1 to entry: A MessagingEndpoint has an Address.

3.70

month

any set of values drawn from the value space of 'month' operation of 'time' datatype, as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.71

MonthDay

any set of values drawn from the value space of 'month' operation of 'time' datatype, as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.72

Operation

active part of an interface where the MessageDefinition sent or received may differ

3.73

Participant

involvement of a BusinessRole in a BusinessTransaction

3.74

Port

communication channel for a participant

3.75

Property

passive part of an Interface with the same MessageDefinition sent and received

3.76

Protocol

kind of message choreography whose Ports provide or require Interfaces with Properties and Operations

3.77

quantity

a counted number of non-monetary units possibly including fractions

3.78

rate

a quantity or amount measured with respect to another measured quantity or amount

EXAMPLE US Dollars per hour, US Dollars per EURO.

3.79

receive

handling of a stimulus passed from a sender instance

3.80

Repository

place where all RepositoryConcepts are stored

3.81

RepositoryConcept

artefact that has been defined during the development of an ISO 20022 MessageDefinition and ResourceDefinition, which is stored in the Repository

3.82

Resource

data set that is managed by an API

Note 1 to entry: Data set maybe expressed as MessageComponent

3.83

ResourceDefinition

formal description of one or more data sets that can be managed by an AP

Note 1 to entry: The ResourceDefinition is built as a tree structure of MessageComponents and DataTypes. A ResourceDefinition is uniquely identified in the BusinessProcessCatalogue.

Note 2 to entry: Like MessageComponents, ResourceDefinitions can be used by various APIs that each have separate market practices

3.84

ResourceSet

set of Resources

3.85

send

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

3.86

string

any set of values drawn from the value space of 'characterstring', as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.87

Syntax

data or interface definition language

Note 1 to entry: A syntax is the specification language of a SyntaxMessageScheme.

EXAMPLE 1 W3C XML Schema 1.0

EXAMPLE 2 ISO ASN.1

EXAMPLE 3 OpenAPI Specification

EXAMPLE 4 RDF SHACL

3.88

SyntaxMessageScheme

specification in a syntax used to define, constrain or validate the structure of a MessageInstance in a possible encoding

EXAMPLE 1 A SyntaxMessageScheme specified in XML Schema validates messages encoded in XML.

EXAMPLE 2 A SyntaxMessageScheme specified in ASN.1 defines the structure of messages encoded as XML, JSON, and several binary formats.

EXAMPLE 3 A SyntaxMessageScheme specified in SHACL constrains RDF graphs encoded in XML, JSON-LD, TTL and several other textual formats.

EXAMPLE 4 A `SyntaxMessageScheme` specified in ISO 3531-1 (FIX tagvalue) defines the structure of messages encoded as tag-value format.

3.89

text

finite array of characters

3.90

time

any set of values drawn from the value space of 'time', as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.91

TopLevelCatalogueEntry

artefact in the `BusinessProcessCatalogue` that is not owned by another artefact in the Repository

3.92

TopLevelDictionaryEntry

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

3.93

trace

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

3.94

TransportMessage

document that is an instance of the `MessageTransportSystem` message schema

3.95

year

any set of values drawn from the value space of 'year' operation of 'time' datatype, as specified by ISO/IEC 11404:2007 General-Purpose Datatypes

3.96

yearMonth

any set of values drawn from the value space of the composite of 'year' and 'month' operations of 'time' datatype, as specified by ISO/IEC 11404:2007 General-Purpose 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 Built-in DataTypes and Enumerations. Further details on these packages can be found in [Annex A](#).

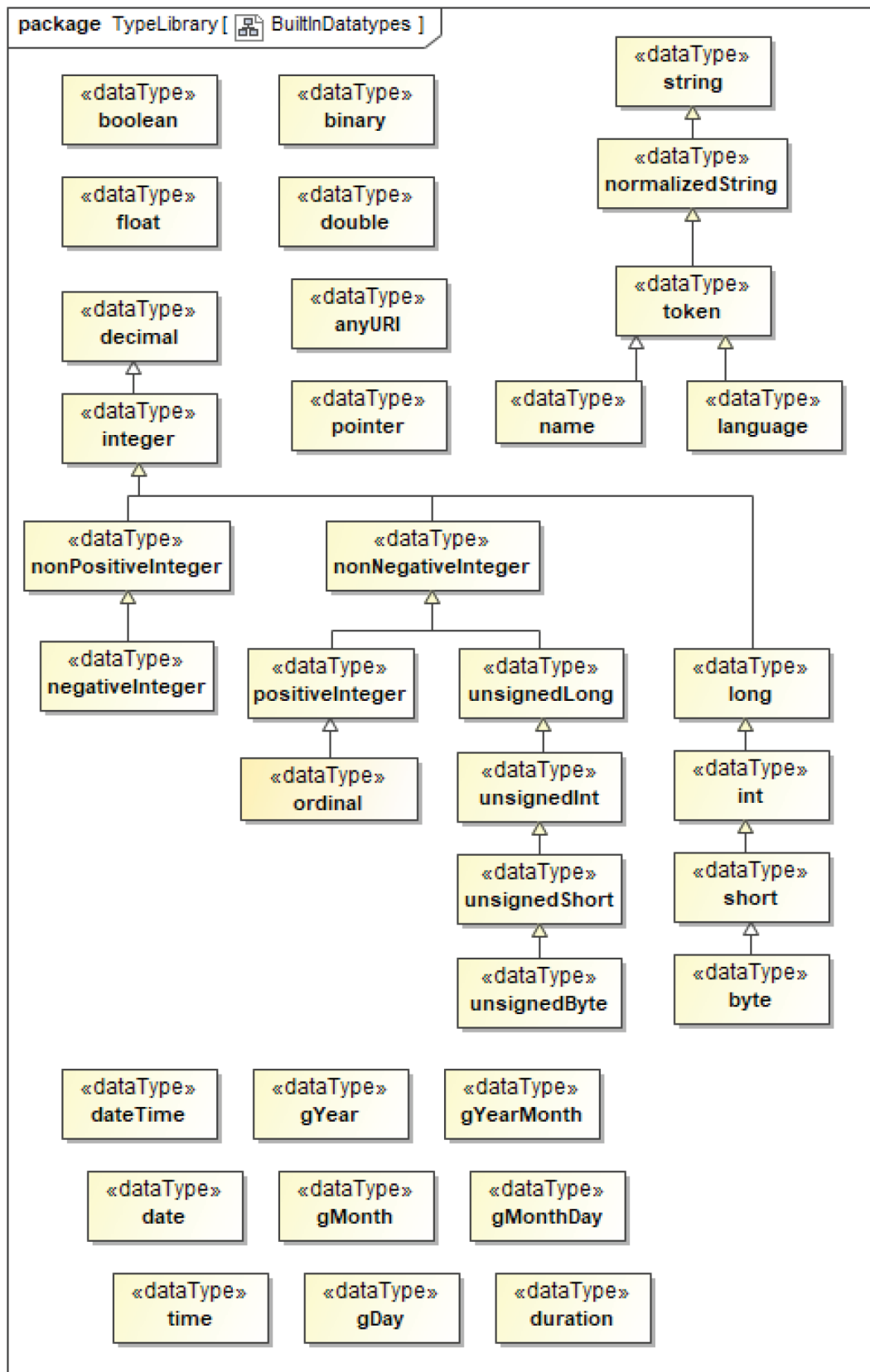


Figure 1 — Type library Built-in DataTypes

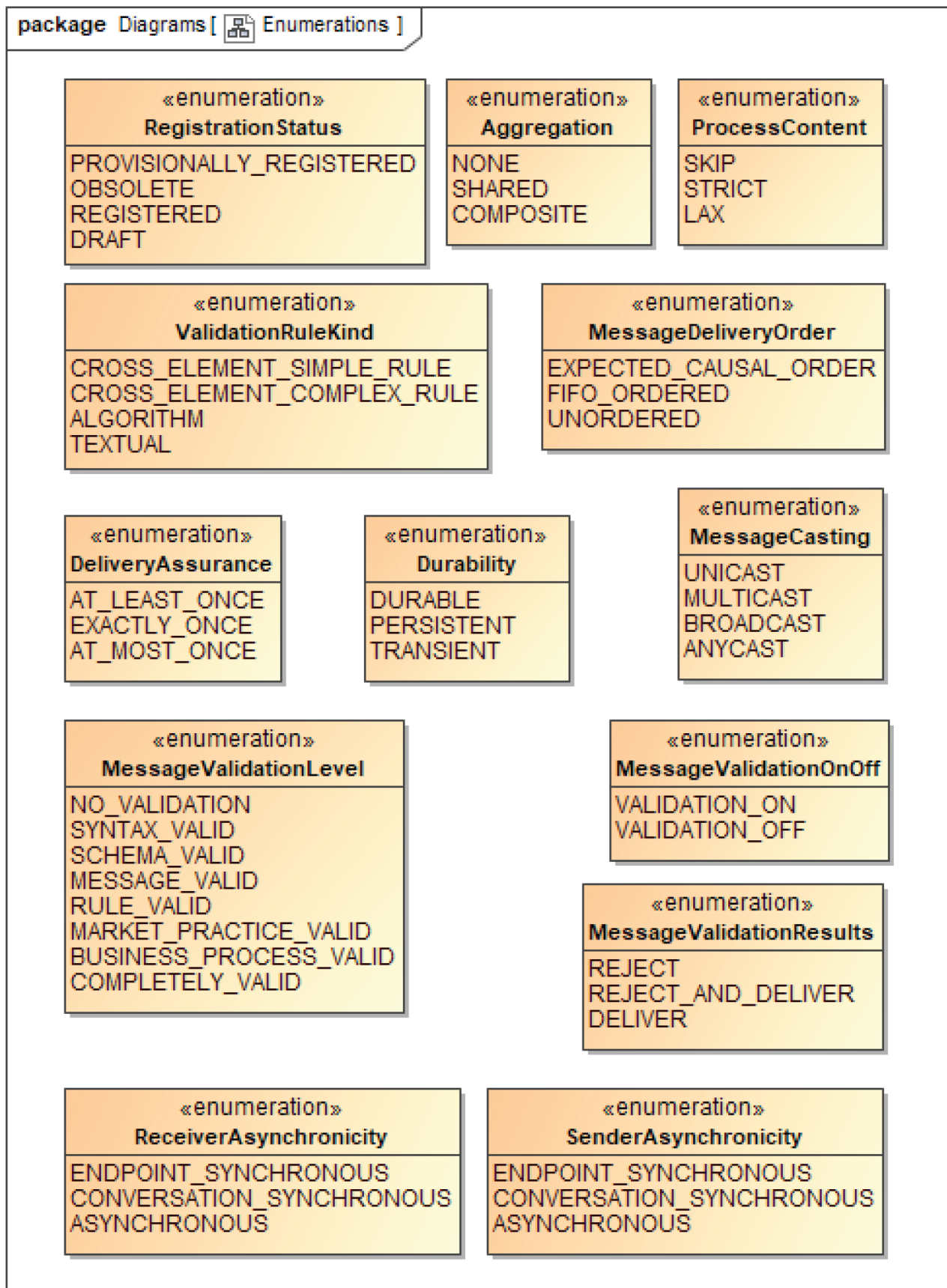


Figure 2 — Type library enumerations

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::BaseTypesPackage contains definitions of the built-in DataTypes in a form that makes it possible for modellers to use these DataTypes in UML and MOF models.

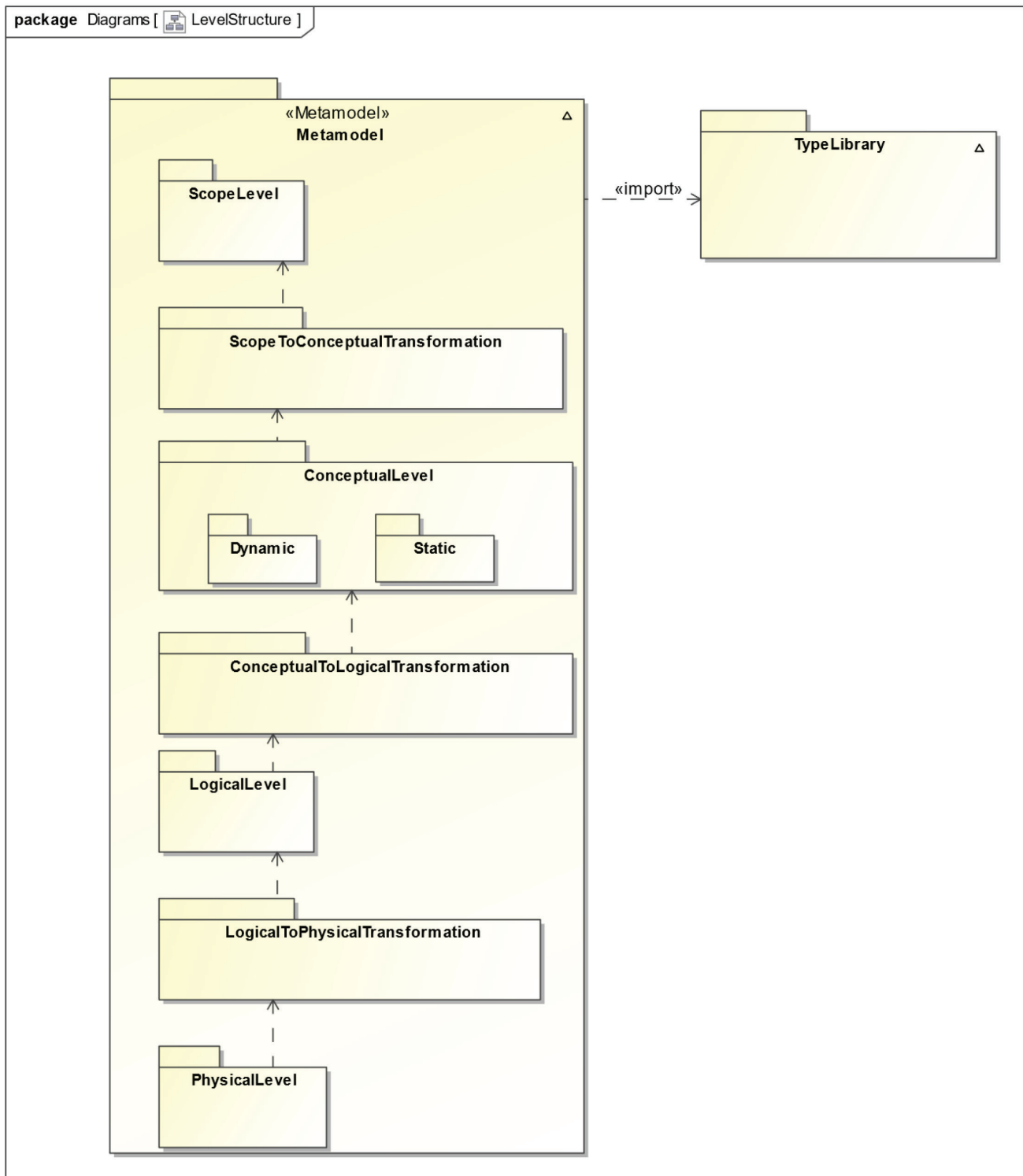


Figure 3 — ISO 20022 level structure

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 Roles in the relevant BusinessProcesses.
Conceptual	Formalizing the semantics and discovering the communication and interaction requirements related to these BusinessProcesses by defining the Participants in BusinessTransactions with their BusinessActivities, MessageTransmissions and BusinessComponents.
Logical	Creating a precise description of the MessageChoreographies comprising 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.

NOTE 3 This does not prelude navigation of the levels via the realisation trace association, for example, from Conceptual to Scope to query associated information.

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

NOTE 4 This might be further constrained in a modelling method.

NOTE 5 The levels do not subsume each other.

NOTE 6 The metaclasses used to explain the different levels are detailed in [Annex B](#).

NOTE 7 ISO 20022-2 - UML Profiles defines a specific UML subset and stereotypes that are used to represent the metamodel when modeling in UML

5.3.1 Scope level

The purpose of the Scope level is to acquire an understanding of the BusinessArea for which an ISO 20022 Conformant BusinessTransaction, MessageSet and API Resources 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.

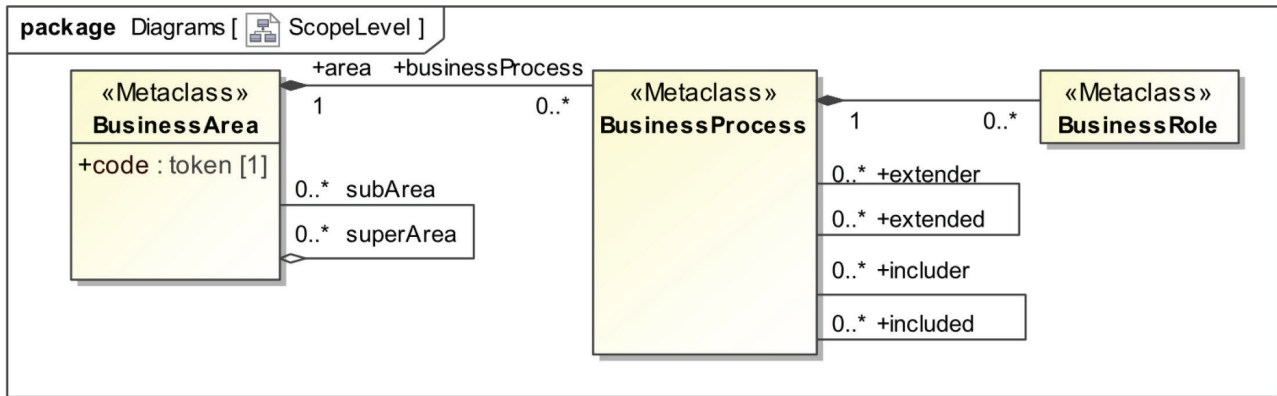


Figure 4 — Scope level

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.

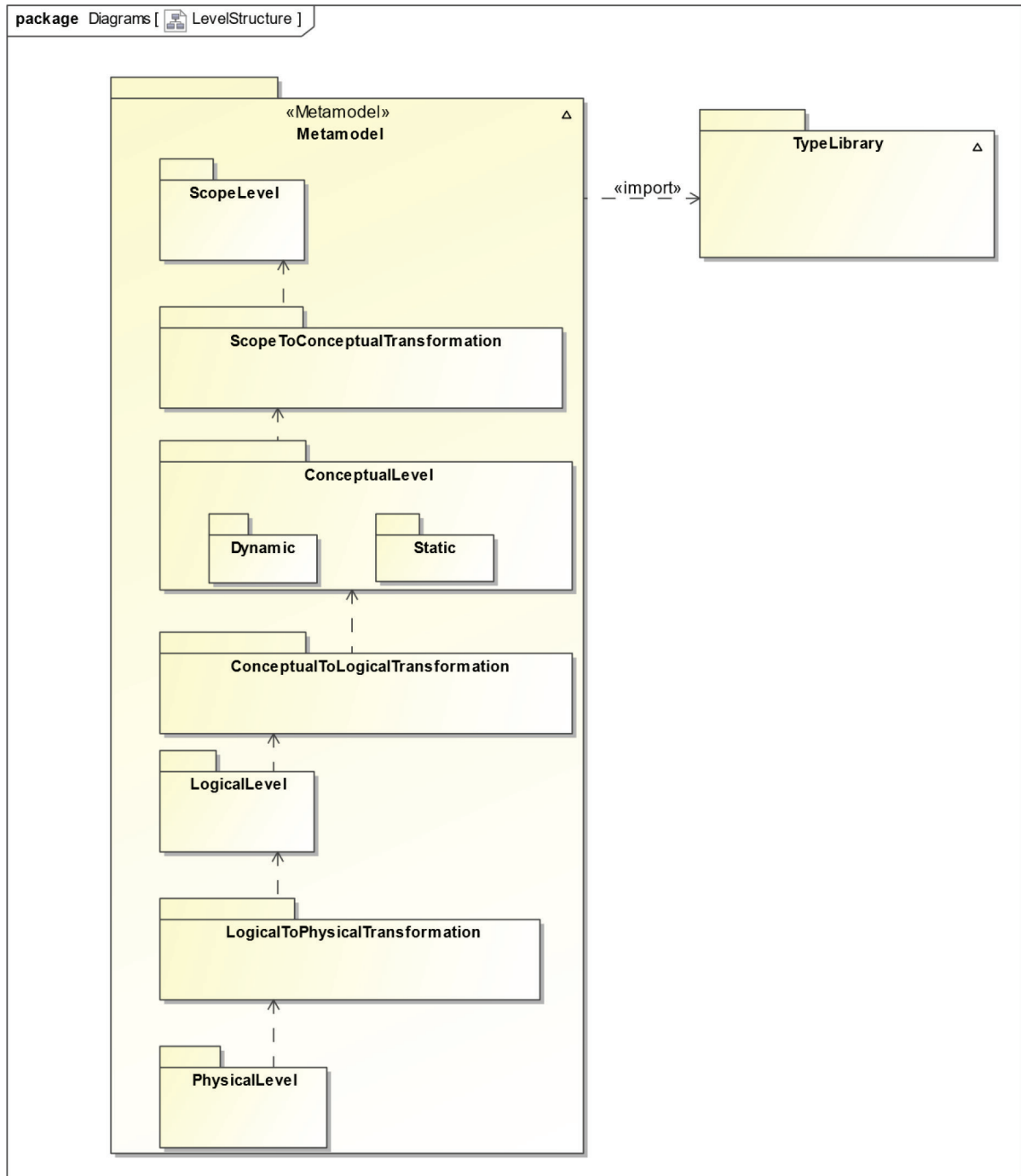


Figure 5 — ConceptualStatic level

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

Define the dynamic part, the process description of the full interaction between all involved Participants, by specifying the details of the BusinessTransaction 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. The focus is now on defining the MessageTransmissions that are needed to get the required information at the right time to the right business user. They are identified by their function in a BusinessArea.

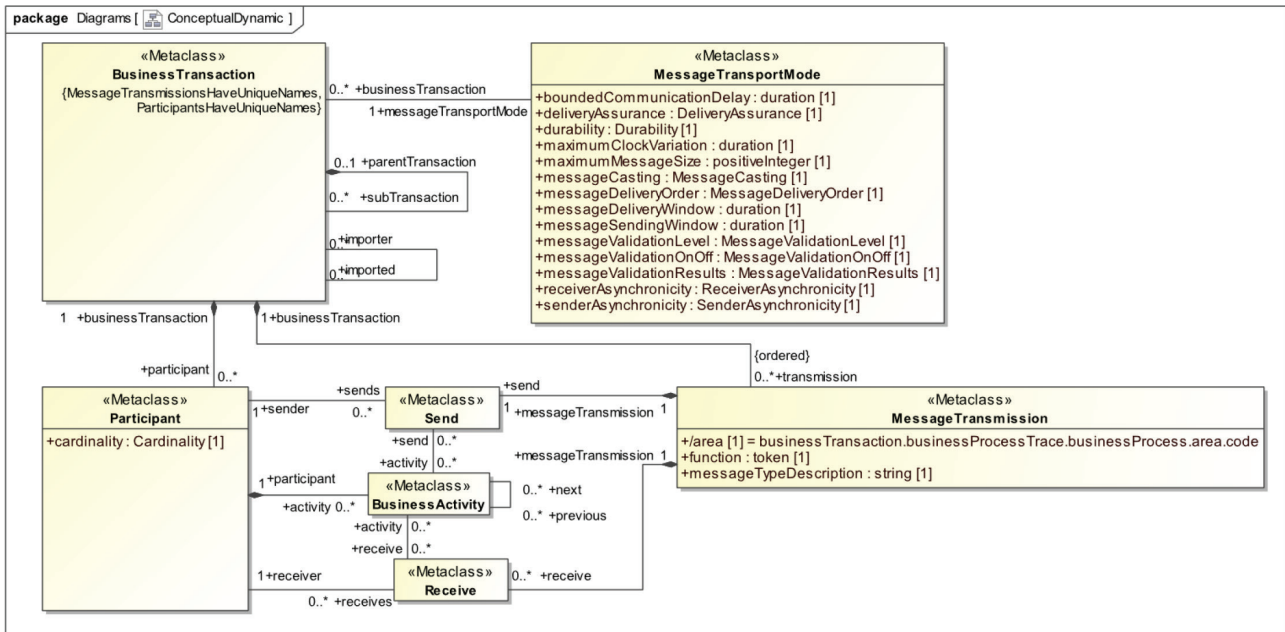


Figure 6 — ConceptualDynamic level

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 of the BusinessTransaction to defining the static part, the precise structure of the MessageInstances, which is done in MessageDefinition.

Each MessageDefinition is a schema, a set of rules for a logical data model of a conceptual MessageTransmission. A MessageDefinitionIdentifier distinguishes message types to which one or several schema may apply. Each MessageDefinition comprises of MessageBuildingBlocks, typed by either reusable DataTypes or MessageComponentType that derive from the BusinessComponent.

There are several kinds of MessageComponentType:

- ExternalSchema – content that conforms with rules defined elsewhere.
- ChoiceComponent – is a list of MessageElement of which only one is permitted.
- MessageComponent – is a list of MessageElement, all of which are permitted.

Each MessageElement is either a MessageAttribute that has a specific DataType or MessageComponentType, or a MessageAssociationEnd referencing a MessageComponentType by composition or reference.

- create ResourceSets with their ResourceDefinitions:
 - the ResourceSet is still characterized from a business perspective;
 - the focus moves from defining the dynamics of the BusinessTransaction to defining the static part, the precise structure of the Resources, which are datamodels managed by API calls

Each MessageElement is either a MessageAttribute that has a specific DataType or MessageComponentType, or a MessageAssociationEnd referencing a MessageComponentType by composition or reference. Each Resource is a represented by a ResourceComponent, together with a set of rules for a logical data model of a conceptual MessageTransmission. A ResourceComponent is a MessageComponent.

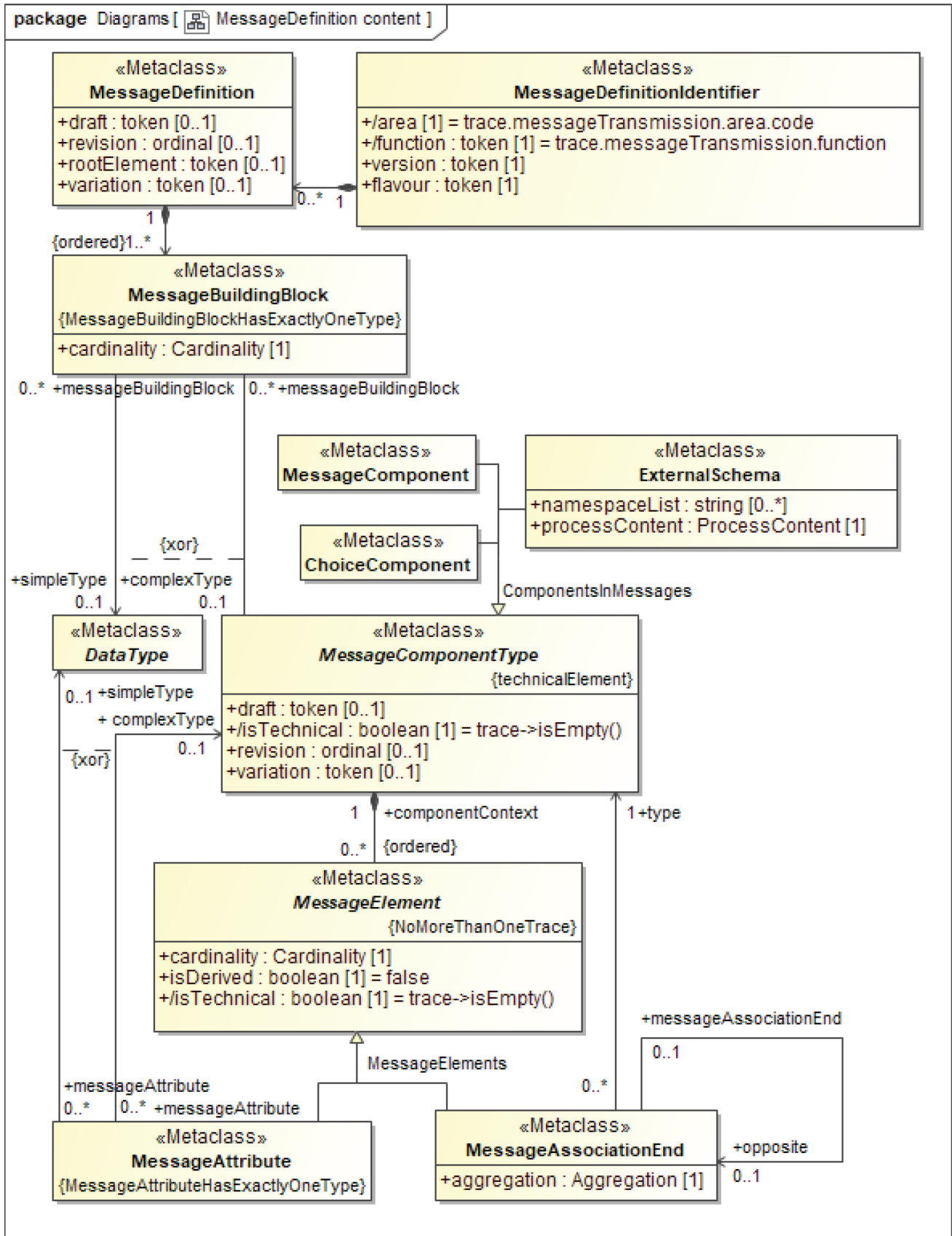


Figure 7 — Logical level - MessageDefinition content

Each MessageDefinition defines a set of rules for messages using a MessageDefintionIdentifier. The MessageDefintionIdentifier further qualifies the BusinessArea and function of a MessageTransmission with a

flavour and version. Each MessageChoreography is a precise and complete description of a MessageInstance exchange within a BusinessTransaction. MessageSet aggregate the MessageDefinitions used in the MessageChoreography. Protocols define Ports which provide or require the Interfaces which specify which MessageDefinitions are used as Properties or as inputs and outputs for Operations.

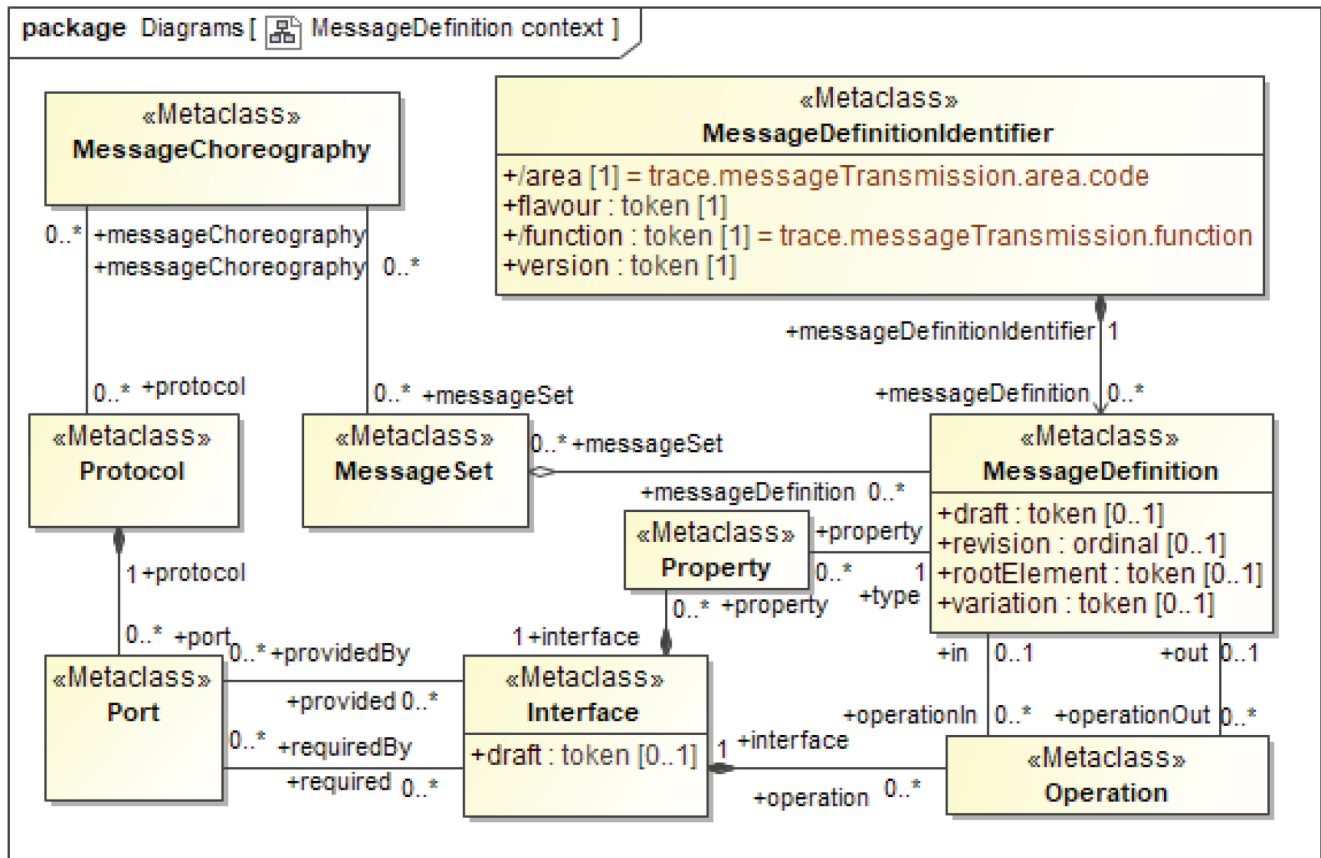


Figure 8 — Logical level - MessageDefinition context

5.3.4 Physical level

The physical level delivers the physical implementation of MessageDefinitions and Constraints in an appropriate syntax, such as XML Schema. 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 Conforming SyntaxMessageSchemes;

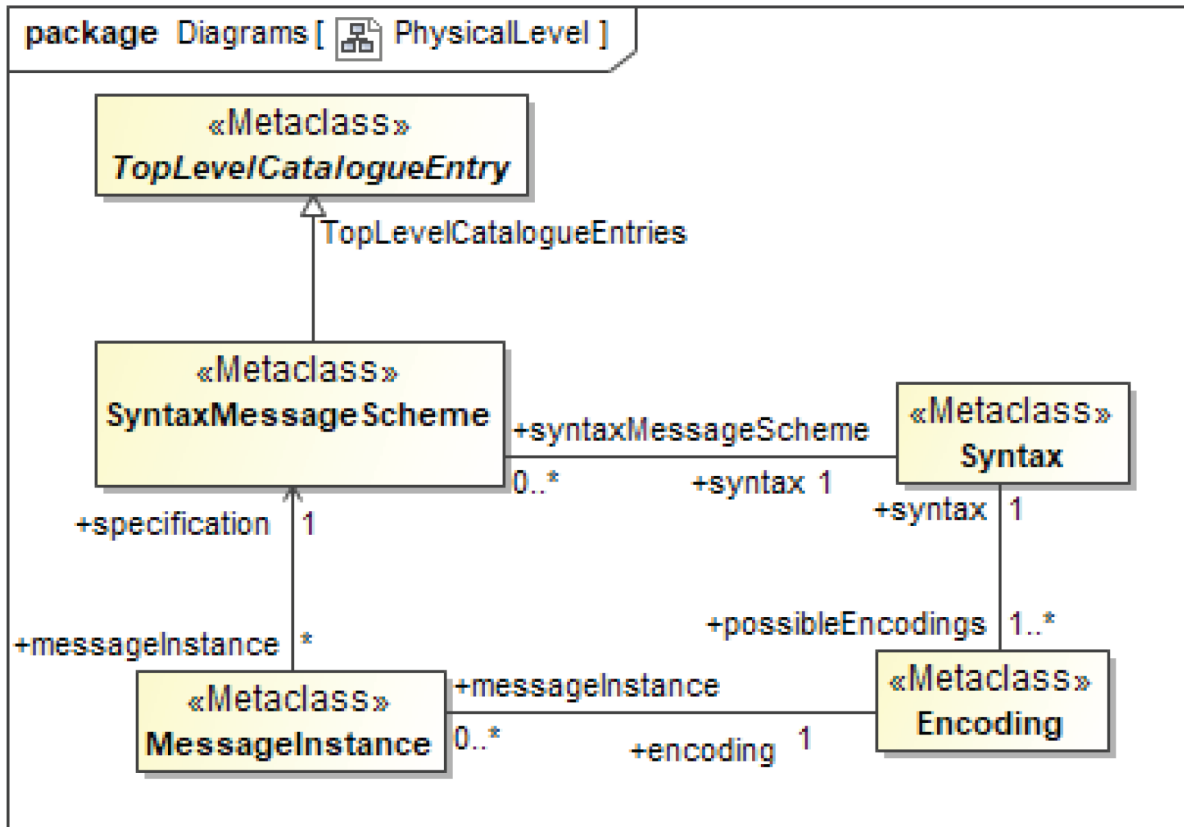


Figure 9 — Physical Level

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.

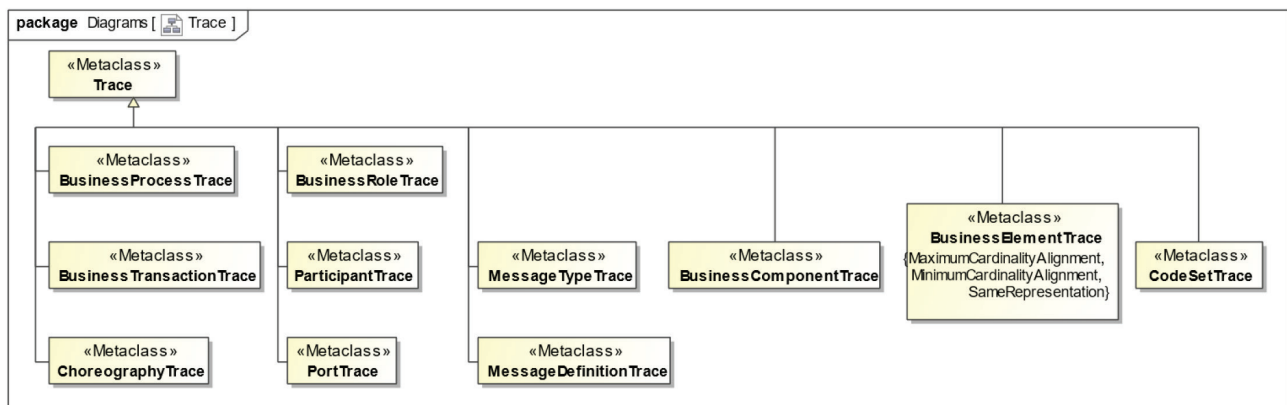


Figure 10 — Traces

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

The Scope level describes the overall communication and all information required to facilitate the communication, without considering how information is communicated between Participants. At the Conceptual level the BusinessProcess executed between the BusinessRoles to enact the BusinessTransactions describes the communication and information needed to satisfy the BusinessProcess activities between 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.

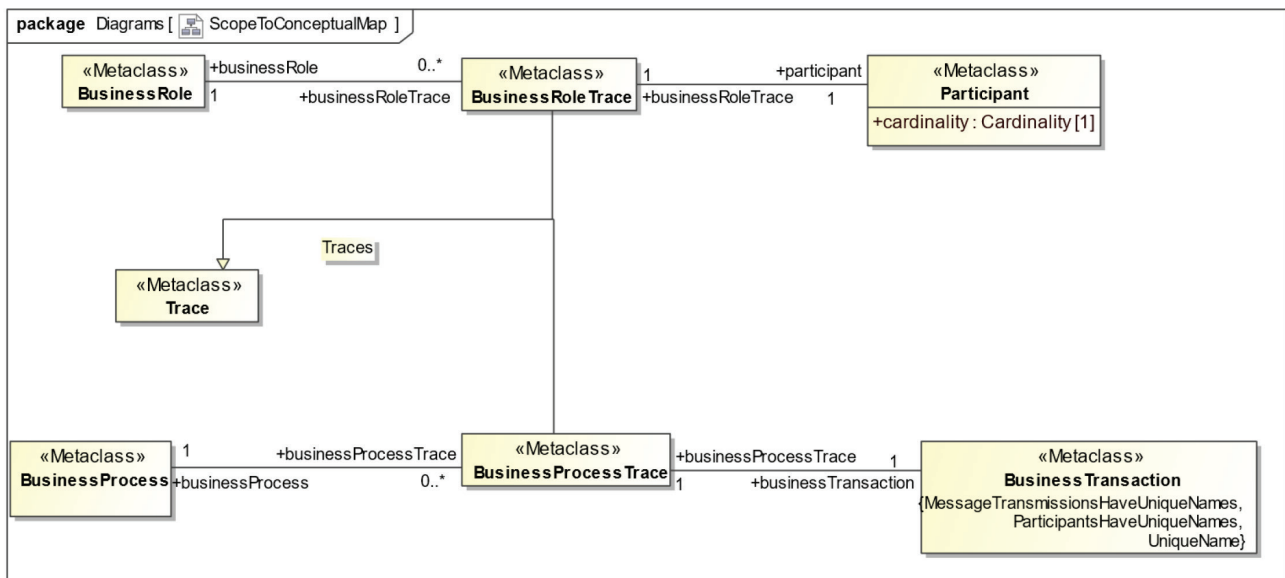


Figure 11 — Scope to Conceptual level mapping

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 trace to MessageDefinitions.

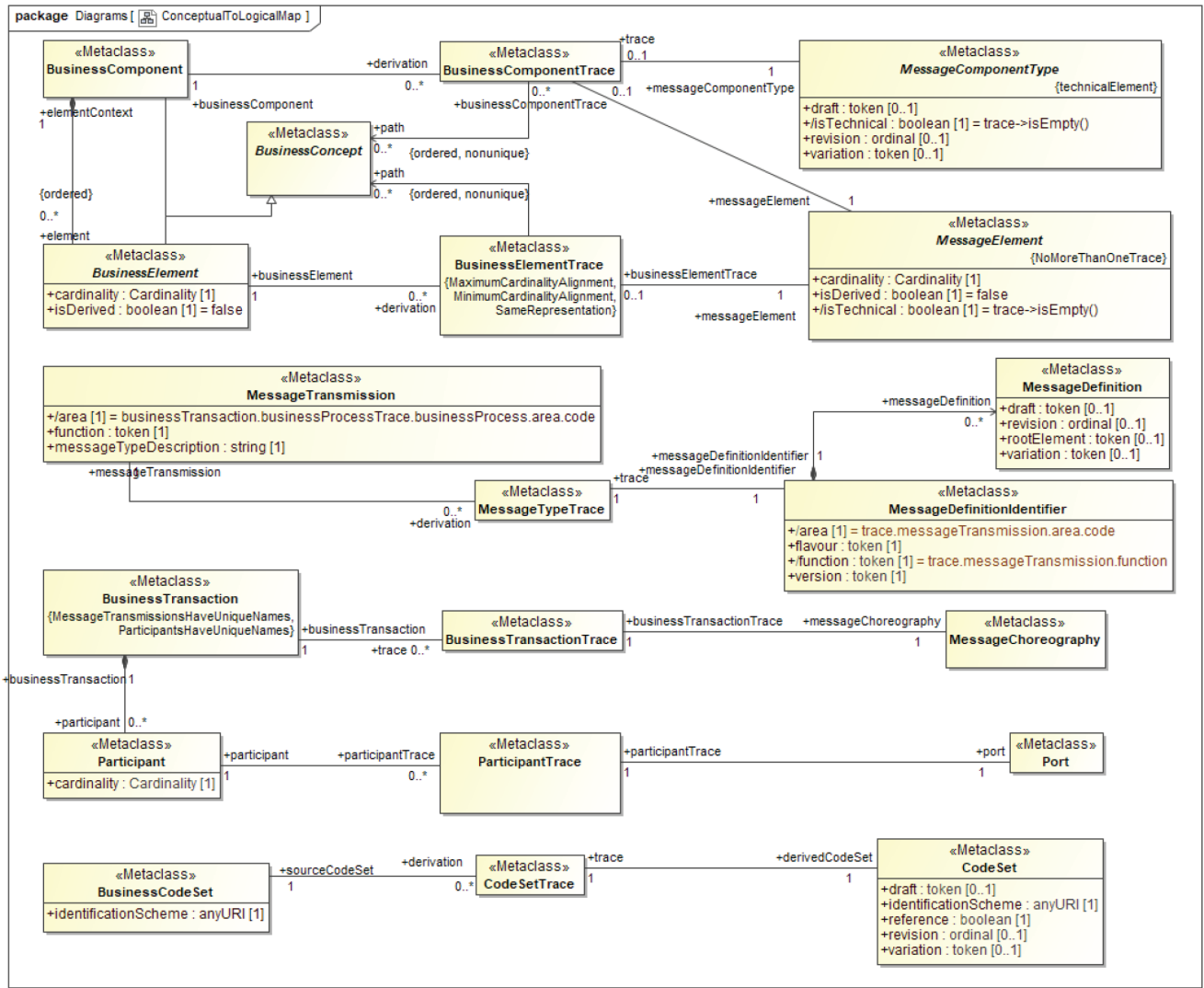


Figure 12 — Conceptual to Logical level mapping

5.4.3 Logical to Physical

This transformation is defined in Part 4 for XML Schema and Part 8 for ASN.1. Part 9 provides general guidelines, requirements and a template for specifying transformation rules from the logical to the physical for other syntaxes or encodings.

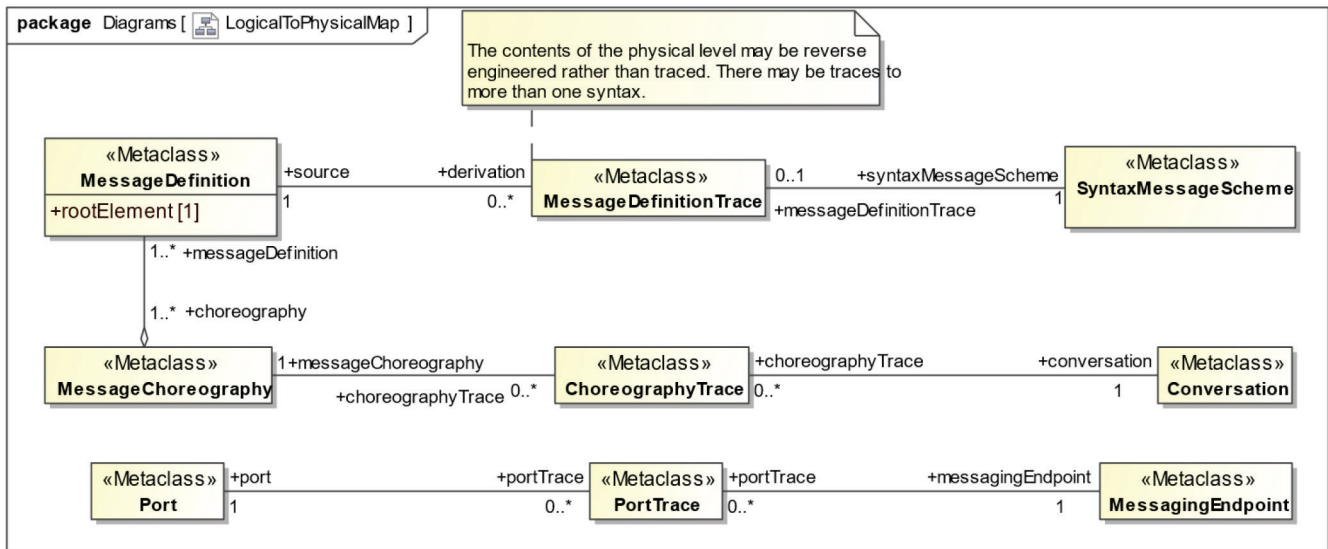


Figure 13 — Logical to Physical level mapping

6 Repository

6.1 ISO 20022 Repository structure

6.1.1 Overview

[Figure 14](#) 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.

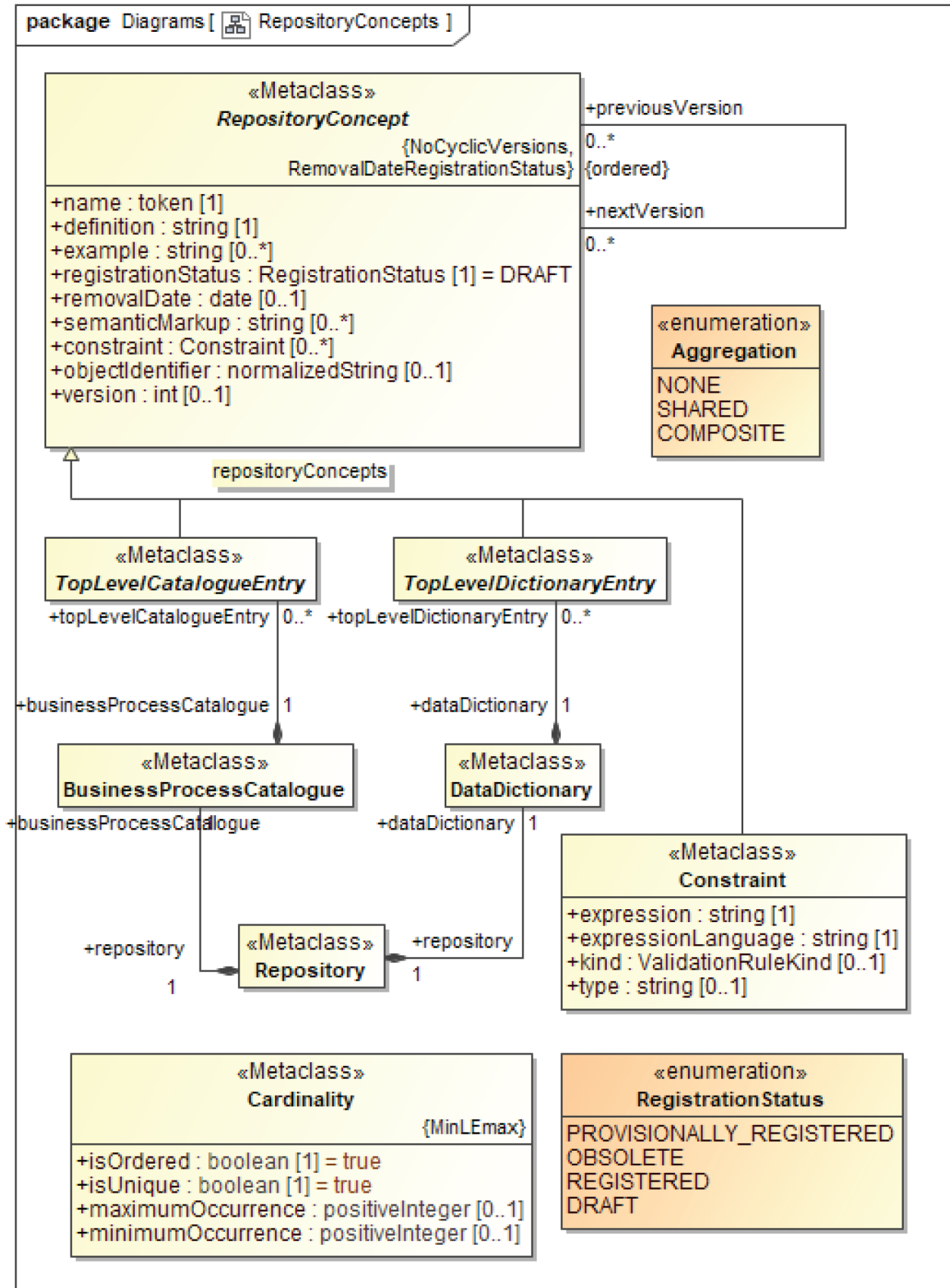


Figure 14 — Global concepts

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. Within the BusinessProcessCatalogue, the TopLevelCatalogueEntries are under release control. A BusinessProcessCatalogue release is always based on one single DataDictionary release.

The main relationships between DictionaryItems and CatalogueItems are as follows.

- Within the DataDictionary: Each MessageConcept derives from a BusinessConcept.
- Within the BusinessProcessCatalogue: Each MessageChoreography derives from a BusinessTransaction, which derives from a BusinessProcess. Each MessageDefinition traces from a MessageTransmission.
- Between the BusinessProcessCatalogue and the DataDictionary: Each MessageDefinition comprises of MessageBuildingBlocks, which have either a simple DataType or a MessageComponentType.

All DictionaryItems and CatalogueItems 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. This contains a reference to an approved business justification or approved Change Request.
- ChangeDate: date at which the change to the item has been registered.

6.2 DataDictionary

6.2.1 Overview

A DataDictionary contains DictionaryItems. Each DictionaryItem 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 DictionaryItems:

- 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.

- a Resource is a MessageComponentType

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 DictionaryItems 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.

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 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 are instances of the DataType metaclass, which do not provide values for the metaproperties. [Figure 16](#) illustrates the allowed instances and the relationship between the datatypes and their metaclass.

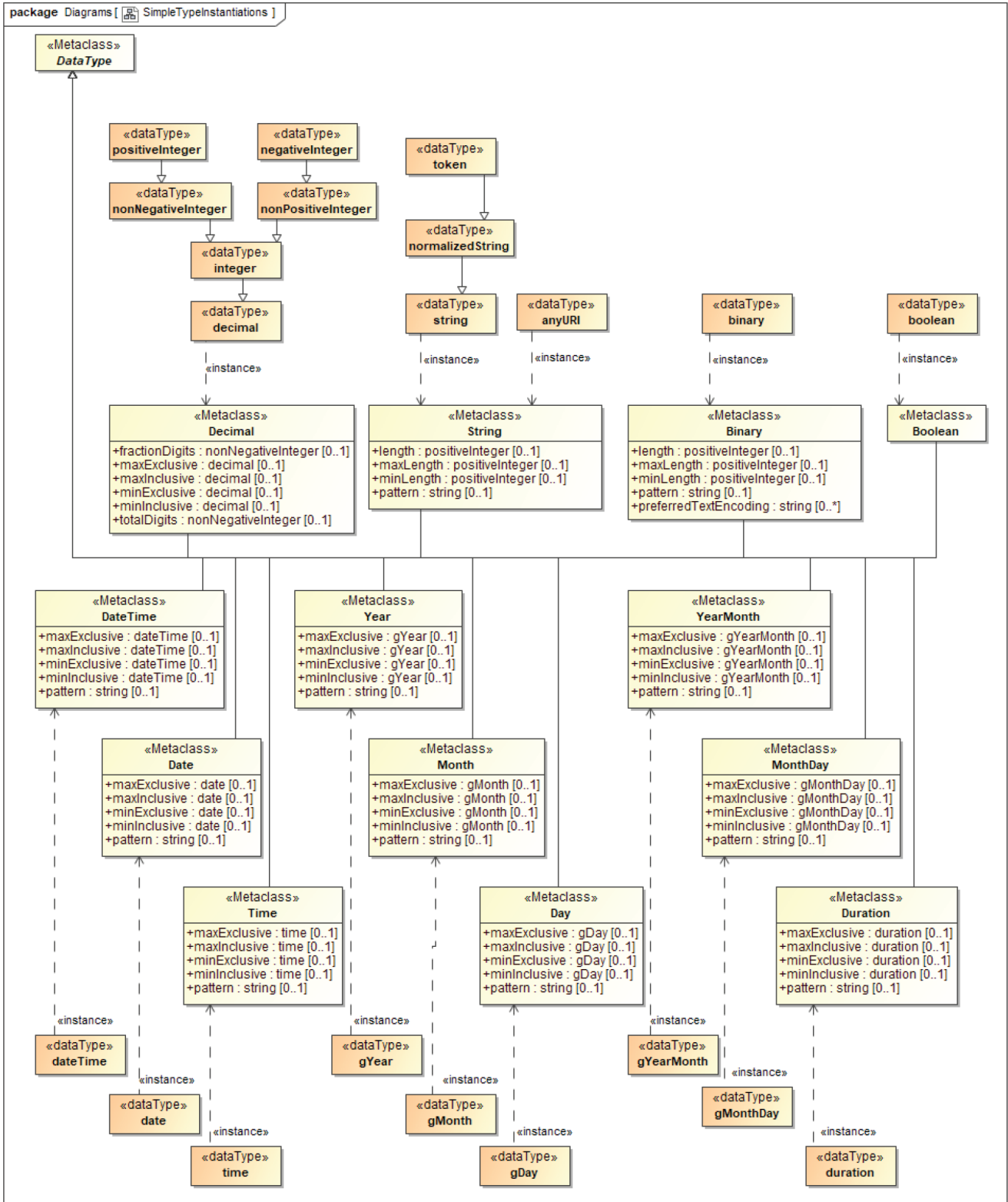


Figure 15 — Simple type instantiations

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.

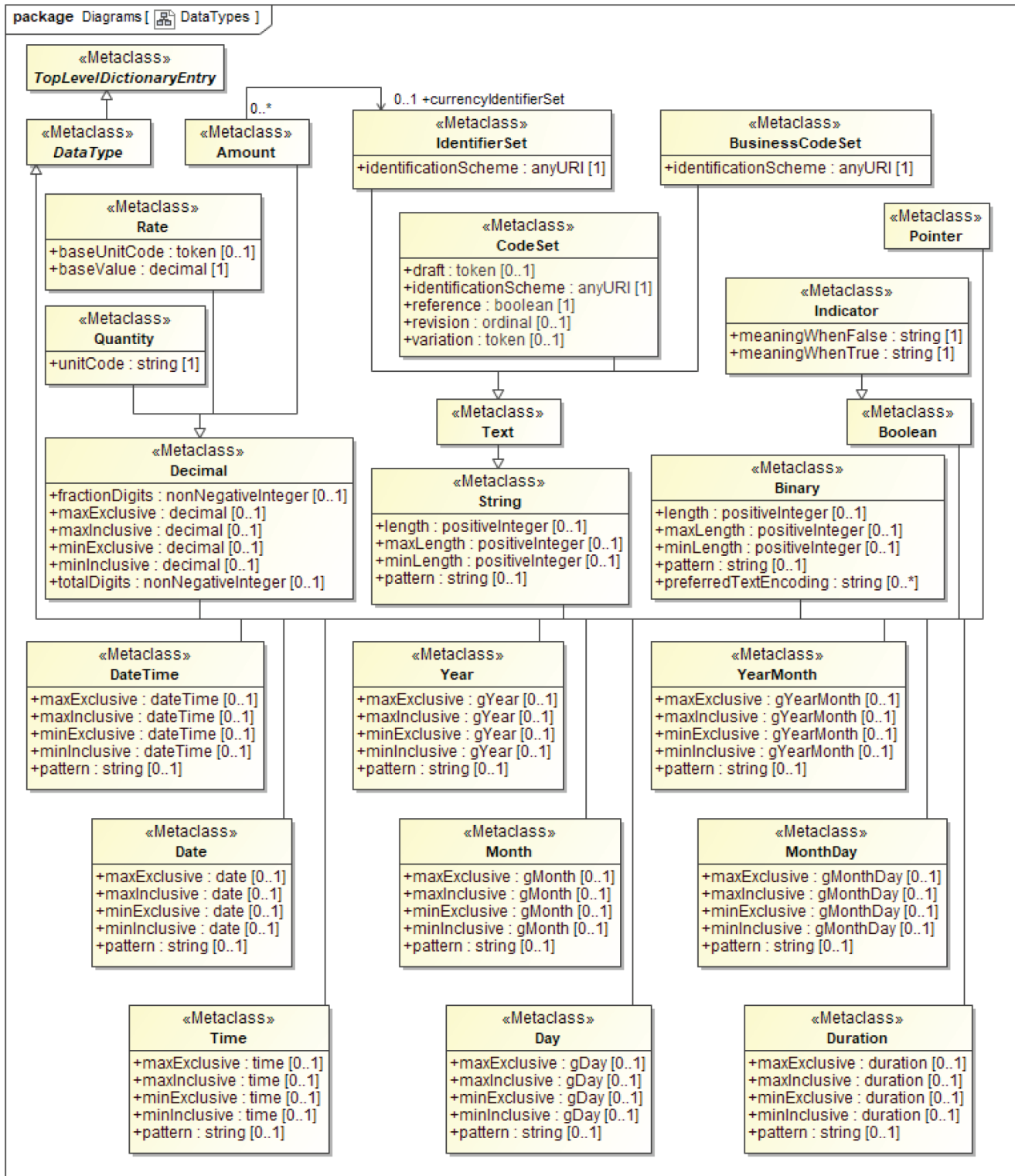


Figure 16 — User-defined DataTypes

Each DataType representation defines the following information.

- a) The ISO/IEC 11404:2007 DataType that will be used for all DataTypes based on this DataType representation, and therefore its value space.

EXAMPLE 1 DataTypes based on DataType representation “Text” will restrict XSD String or JSON String

EXAMPLE 2 DataTypes based on DataType representation “Amount” will restrict XSD Decimal or JSON number.

- 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 DictionaryItems that are defined and used during the Logical level activities. They form the basic items of MessageDefinitions.

- MessageComponentTypes are the re-usable DictionaryItems 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 (expressed through a path that traverses the business model identifying those BusinessComponents that contextualise the MessageComponentType) 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 DictionaryItem Registration Status

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

- **DRAFT:** the DictionaryItem is in development and may change at any time. This enables submitters who iterate their design whether internally or in collaboration with the Registration Authority.

- **PROVISIONALLY_REGISTERED**: the DictionaryItem is pending final approval (as defined in ISO 20022-7). This enables the Registration Authority to inform a community of users of DictionaryItems that the item will potentially become “REGISTERED” in the near future.
- **REGISTERED**: the ISO 20022 Conformant DictionaryItem is approved (as defined in ISO 20022-1) and can be used.
- **OBSOLETE** [**<REMOVAL-DATE>**]: the DictionaryItem is no longer considered as ISO 20022 Conformant (as defined in ISO 20022-1) and may no longer be used for registering updates to the ISO 20022 Repository. The DictionaryItem will, however, be kept in the DataDictionary as long as it is used in other DictionaryItems or CatalogueItems. The optional **<REMOVAL-DATE>** enables the Registration Authority to inform a community of users that the DictionaryItem will be physically removed from the DataDictionary at the date specified as **<REMOVAL-DATE>**. An obsolete DictionaryItem without **<REMOVAL-DATE>** indicates an item has been “deprecated” and therefore may no longer be used.

6.2.4 DictionaryItems description information

The semantics of each DictionaryItem is defined by the following information¹⁾;

- **Name**: official name of the DictionaryItem (uniqueness is defined in the metamodel). It may include versioning information.
- **Definition**: precise description of the meaning of the DictionaryItem.
- **Removal date**: the date at which a DictionaryItem 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 DictionaryItem in its business context.
- **Multiplicity**: indicates whether the DictionaryItem 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

The published DataDictionary contains all DictionaryItems, i.e. with status **PROVISIONALLY_REGISTERED**, **REGISTERED** or **OBSOLETE** [**<REMOVAL-DATE>**]²⁾. This allows the Registration Authority to inform communities of users of DictionaryItems that will either become registered, or will be removed from the Dictionary in the near future. The process of publishing and historical record keeping of the DataDictionary is described in ISO 20022-7.

6.3 BusinessProcessCatalogue

6.3.1 Overview

The BusinessProcessCatalogue contains BusinessAreas. A BusinessArea can be described in more details using BusinessProcesses. The communication requirements and the interaction requirements in the various BusinessAreas and BusinessProcesses 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.

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.

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 BusinessProcessCatalogueItems

The following items may be contained in the BusinessProcessCatalogue:

- BusinessArea with its MessageDefinitions and their MessageBuildingBlock (Diagram);
- BusinessProcess (Diagram) with its BusinessRoles;
- BusinessTransaction (Diagram) with its Participants;
- MessageSet with its MessageDefinitions (Diagram);
- SyntaxMessageScheme;
- MessageChoreography;
- MessageTransportMode.

6.3.3 CatalogueItem Registration Status

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

- PROVISIONALLY_REGISTERED: the CatalogueItem is pending final approval (see ISO 20022-7). Note that, unlike for DataDictionaryItems, CatalogueItems with this status are not made publicly available;
- REGISTERED: the ISO 20022 Conformant CatalogueItem is approved;
- OBSOLETE [<REMOVAL-DATE>]: the CatalogueItem is no longer considered as ISO 20022 Conformant and may no longer be used for registering updates to the ISO 20022 Repository. The CatalogueItem 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 CatalogueItem will be physically removed from the BusinessProcessCatalogue at the date specified as <REMOVAL-DATE>.

6.3.4 CatalogueItem description information

The semantics of each CatalogueItem is defined by the following information³⁾:

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

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.

6.3.5 BusinessProcessCatalogue life cycle

The Registration Authority makes publicly available the current release of the BusinessProcessCatalogue. This published BusinessProcessCatalogue contains all CatalogueItems with status `REGISTERED` or `OBSOLETE` [`<REMOVAL-DATE>`]. CatalogueItems that have the status `<PROVISIONALLY_REGISTERED>` will not be accessible via the public BusinessProcessCatalogue. The process of publishing, historical recording keeping and access to interim versions of the BusinessProcessCatalogue is covered in ISO 20022-7.

7 Registration

Submission requests for updates to the DataDictionary or BusinessProcessCatalogue will include new or updated BusinessProcesses, BusinessTransactions and MessageDefinitions. The models shall be Conformant with the metamodel and, if submitted in UML, with ISO 20022-2 and with ISO 20022-3. New or updated Repository items shall be identified.

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

8 Repository Access

8.1 General

The ISO 20022 Repository information is to be made publically accessible. 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

Access to the ISO 20022 Repository information shall include the following output types:

- a) Interactive search/queries on:
 - the DataDictionary, giving access to all DictionaryItems and exploiting the relations between DictionaryItems;
 - the BusinessProcessCatalogue, giving access to all CatalogueItems and exploiting the relations between CatalogueItems.
- b) A set of predefined downloadable extracts providing information about:
 - the DataDictionary, listing all DictionaryItems with all relevant information, e.g. Name, Definition, etc.;
 - the BusinessProcessCatalogue (BusinessProcesses 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 on the ISO 20022 website.

The downloadable extracts shall be provided free of charge via the ISO 20022 website in the formats described in ISO 20022-7.

Repository extracts in XMI and may be requested on a case-by-case basis, to deliver ISO 20022 Repository information in other structured languages or processable formats.

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 2.5.1 and is expressed in XMI 2.5.1.

The content of [Annex A](#) is generated from the metamodel file (currently a MagicDraw file). Changes must be made to that file for generation. The meta-model and template files are available on the ISO standards maintenance portal.

A.2 Type Library Details

A.2.1 Package ISO20022::TypeLibrary

Applied Stereotype	
--------------------	--

Description

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. 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. 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 may 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 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 NOTE If the TransportMessage cannot be delivered then the sending Messaging Endpoint must 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 NOTE The MessageTransportSystem does not deliver notification errors for non-delivery. If the TransportMessage cannot be delivered then the sending MessagingEndpoint must 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 NOTE 1 The first MessagingEndpoint to consume the TransportMessage causes it to be removed from all other MessagingEndpoints. 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. Example 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. Example 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 must 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 must be validated

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

Owned Literals

Literal name	Description
SKIP	the contentmodel must not be validated
STRICT	the contentmodel must be validated against its specification
LAX	the contentmodel must 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 must 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. 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
DRAFT	The DictionaryItem is in development and may change at any time. This enables submitters who iterate their design whether internally or in collaboration with the Registration Authority.

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 must not block the sending or receipt of other TransportMessages while waiting for a response to the sent TransportMessage

A.2.2.13 Enumeration ValidationRuleKind**Description**

Indicates the rule is descriptive, calls out to specific implementation, or is cross element with or without precondition.

Qualified Name	ISO20022::TypeLibrary::Enumerations::ValidationRuleKind
Visibility	public

Owned Literals