EC-Design

The Enterprise Application Integration Standard

Web Services and Information Exchanges designed the easy way

www.ec-design.nl

This page intentionally left blank.

Overview of EC-Design ®

A critical part of engineering information exchanges is to completely understand exactly what functionality is desired. One of the problems is that the people who really understand the domain, the business process experts, do not understand the technology. EC-Design introduces an intuitive way to model functional business requirements and then generate technical specifications to implement them. EC-Design thus realizes a way to specify information exchanges without any knowledge of the underlying implementation technology.

EC-Design provides expert functionality for application integration based on electronic messages and web services. EC-Design is a powerful UML (Unified Modeling Language) data- and message modeling tool which enables you to specify any number of messages or web services in a consistent manner, based on one shared enterprise / business data model.

EC-Design supports the design and implementation of an information architecture based on electronic messages or web services. EC-Design complies with the ISO standard methodologies 11179 and 17113 for data model design and electronic message specification. Both functional and technical message specifications are supported. XML Schema's, JSON Schema's, Edifact mappings, Inhouse file formats as well as RTF and HTML documentation and sample messages with understandable data, can be generated automatically from functional specifications.

EC-Design will speed up and at the same time enhance the quality of engineering and implementing application integration based on electronic messages and web services.

EC-DESIGN WILL SAVE TIME AND MONEY, DURING DESIGN, IMPLEMENTATION AND MAINTENANCE.

EC-Design is designed for professional use by information architects and business process engineers. Single user and multi-user licenses are available commercially. EC-Design is developed and tested for Microsoft Windows operating systems.

ISO standard 11179

ISO 11179 (Specification and standardization of data elements) provides a way to reuse a data element that meets a need, or to design a new data element if one does not already exist. Even before data elements are used in practice, information architects and business process engineers must have a way to identify and describe data logically so that they do not inadvertently introduce inconsistent values of data. If they are to create products that share data, they must first be aware whether or not a data element with the required characteristics already exists. If it does, they should use it, if it doesn't, they should create it.

ISO 11179 describes a data element registry to assist users of shared data to have a common understanding of a data elements meaning, representation, and identification. ISO 11179 aids in the development of precise descriptions of data elements. Data elements that have been formulated according to the principles of ISO 11179 enable interchangeability and retrieval regardless of the information processing system or telecommunication protocols employed.

ISO 11179 provides means to assure data coherence. A registry can serve software developers by enabling the consistent use of data throughout the Software Development Life Cycle (SDLC). A registry will provide the mechanisms for managing data elements and for ensuring their traceability between SDLC phases. For developers of a data dictionary, data element registry, CASE tool, and other data management software, ISO 11179 provides the basis for designing a meta model necessary to enable the capture, storage, management, and exchange of the data element metadata.

ISO standard 17113

The ISO 17113 method for the development of messages defines a process for developing comprehensive, interoperable and certifiable information exchange among independent information systems.

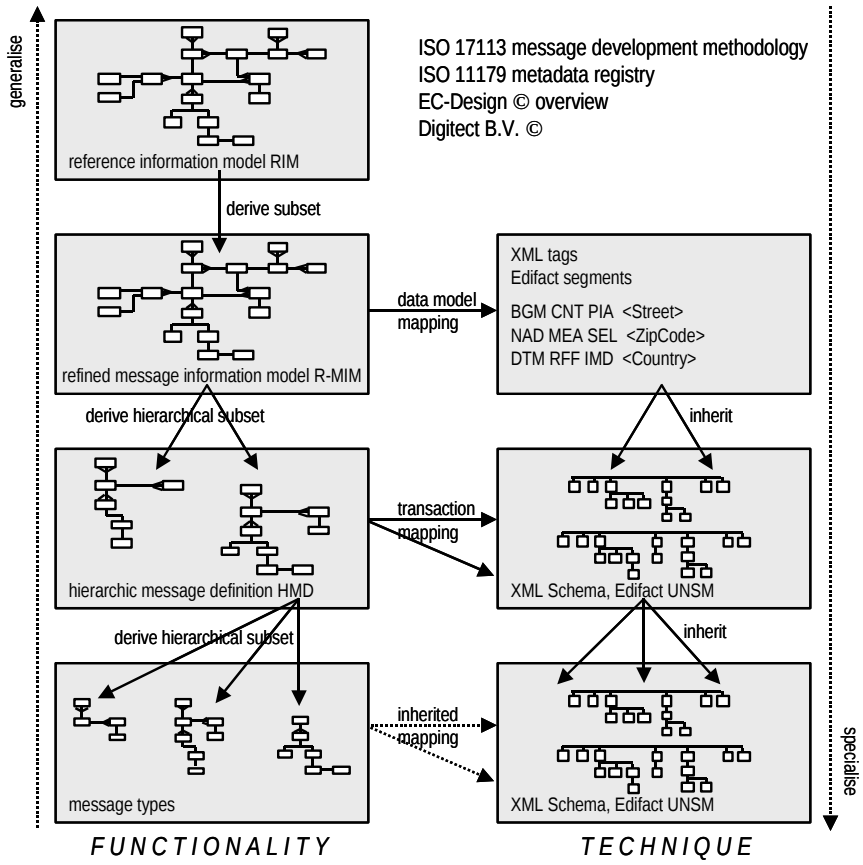
ISO 17113 recognizes three types of information models during the message modeling process. Each information model consists of classes, their attributes, and the relationships between classes; state-transition models; and data types and constraints. The components are defined in a meta model, and the textual representations are maintained in a database. Graphical representations are maintained using UML.

- The first information model that is recognized is the domain information model (DIM), which expresses the information content for a specific area of expertise or interest. The DIM represents one group's view of the world. The harmonization of all DIMs leads to the global Reference Information Model.
- The second model is the Reference Information Model (RIM). The RIM is a coherent, shared information model that is the basis for the semantic specification of information exchanges. All messages and related activities must be derived from the RIM, thus forming the source for the data content for all derived messages. The RIM, by nature, must be high level and generic. The RIM, in its most generic form, models entities and acts, along with roles, relationships and participations.
- The third model is the Refined Message Information Model (R-MIM) that takes the generic RIM and defines a constrained subset that deals with a specific set of events. The R-MIM is used to express the information content for one or more related messages. The R-MIM permits a more specialized information model and supports message specific information constraints.

Besides the information models mentioned above, ISO 17113 recognizes the Hierarchical Message Description (HMD), which specifies a set of messages based on one R-MIM. Each Message Type is specified in exactly one HMD, one HMD may specify several Message Types.

Finally, ISO 17113 introduces the interaction model (IM), which describes the parties that exchange messages and the interactions between those parties. The IM specifies the information flows and makes it possible to structure conformance claims. An interaction defines a specific instance of information exchange. It specifies the trigger event, the message content, and the responsibilities of the receiver. Each interaction is supported by a message definition (HMD). There must be an interaction for each trigger event, and there can be more than one interaction. An interaction sequence is a time order sequence of relevant events. Sequence diagrams are used to document these sequences and are derived from the scenario descriptions.

The following figure depicts the relationship among all components mentioned previously. It also illustrates the way an R-MIM is supported in EC-Design by means of a *data model*, a HMD is supported in EC-Design by means of a *transaction*, and a Message Type is supported in EC-Design by means of a *functional message*.



The process for creating a message as defined by ISO 17113 is to create the RIM, the R-MIM, and build the HMD. The R-MIM will determine the order and hierarchy of the message element types. Actual message formats are then defined by adding additional constraints to the HMD. The HMD includes the information model mapping, the message elements, common constraints and defaults, and the individual Message Type definitions. This process is fully supported by EC-Design.

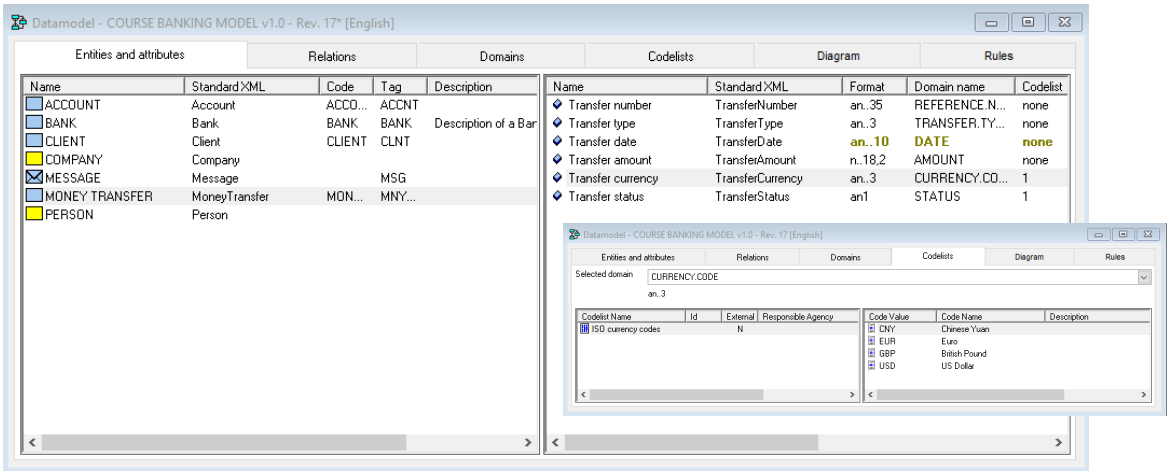
Zoom in on EC-Design ®

EC-Design is based on and fully complies with the ISO standard methodologies 11179 and 17113 for data model design and electronic message specification. EC-Design approaches the message modeling process from a functional point of view, which means that first the logical / semantic business objectives and requirements regarding information exchange, are analyzed and modeled in EC-Design. Only when these logical / semantic objectives and requirements are modeled to their full extend and in every possible detail, the step towards the implementation technique is taken.

Modeling functionality for information exchanges in a certain business area is done in three consecutive steps:

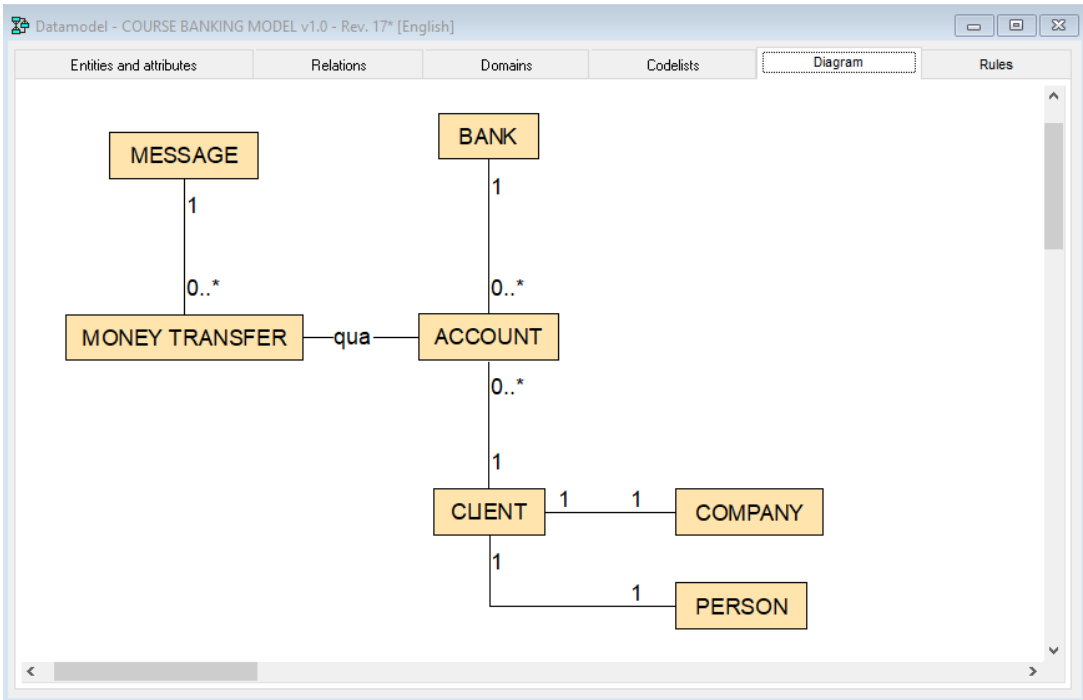
1

- First the *data model* is defined, which describes in detail every possible information object (class, entity, attribute, domain, data type, relationship, …) that is relevant to the exchange of information between the actors in the business area of interest. Semantic rules and constraints that are applicable to the information objects regardless of their specific use in a message are part of the data model and therefore should be modeled here.



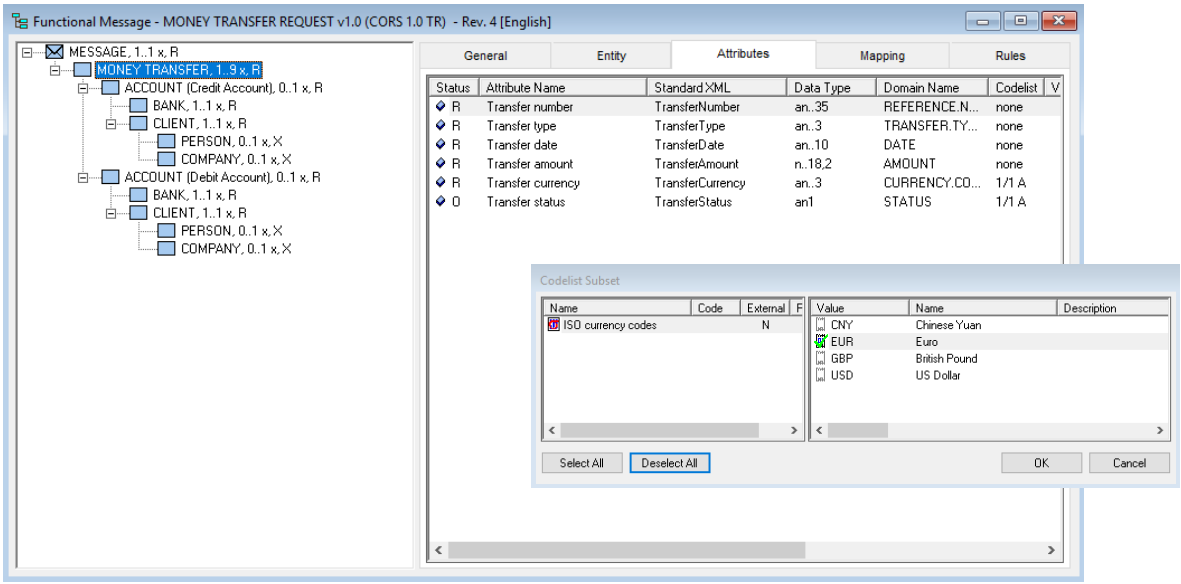
2

- Second the *transaction* is defined as a hierarchical subset of the data model. The transaction contains references to all information objects in the data model that are relevant to a specific process within the business area of interest. Since the transaction is modeled as a hierarchy, information objects can be referenced more than once. Each reference to an information object in the data model can have its own specific rules and constraints. These rules and constraints are not applicable to the data model in general, but only to the transaction and all its related functional messages. Apart from that, of course, the rules and constraints identified in the data model, also apply to the transaction that originates from it.



3

- Third the *functional message* is defined as a hierarchical subset of the transaction it originates from. The functional message may only contain references to information objects in the transaction that are relevant to the message type the functional message defines. Each reference to an information object in the transaction can have its own specific rules and constraints. These rules and constraints are not applicable to the transaction in general, but only to the functional message at hand. Apart from that, of course, the rules and constraints identified in the data model and the transaction, also apply to the functional message that originates from them.

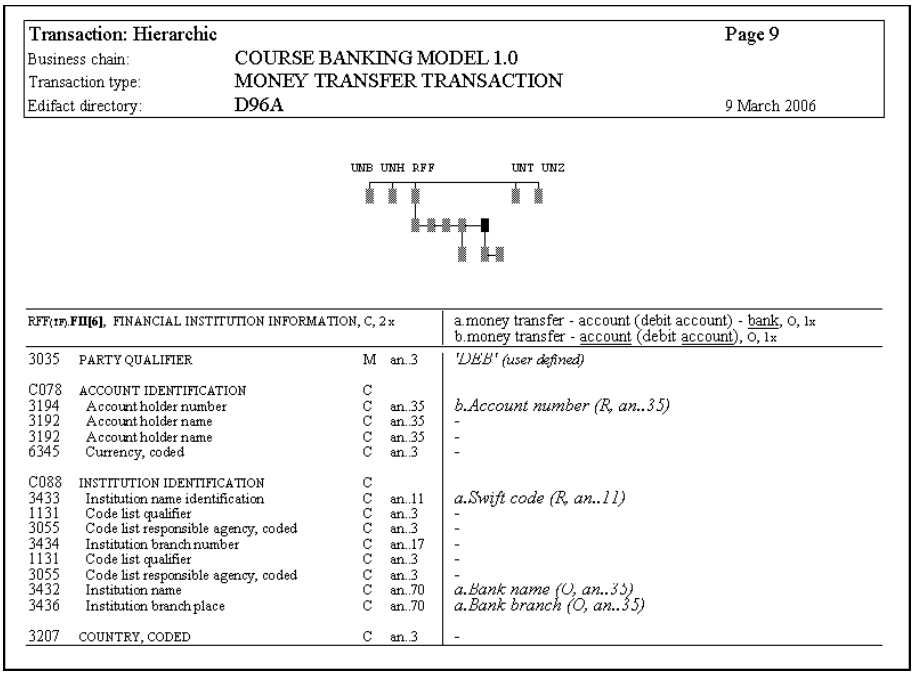


Some examples of rules and constraints that can be set on data model, transaction or functional message level:

- A possible constraint that can be defined in a data model is to restrict the possible values for an attribute to the codes in a code list. This constraint can then be refined in a functional message definition, to restrict the possible values for an attribute to a subset of codes from the code list (set of codes) defined in the related data model.
- In a data model the relationship between two entities can be defined as *1:n* (one-to-many). This relationship can then be used in a transaction hierarchy and a functional message hierarchy. On transaction or functional message level, the relationship can be refined to *1:1* (one-to-one).
- In a transaction the status of an attribute can be set to *O* (optional). In a related functional message the status for this attribute can then be set to *R* (required).

Based on the modeled definitions of functional messages, transactions and data models, EC-Design can generate both functional and technical documentation. Which forms a consistent, coherent, unambiguous and complete set of specifications that is input to the implementation phase, in which the modeled information exchanges are actually built. EC-Design supports output formats like RTF, HTML, XML, XML Schema, JSON, JSON Schema, Edifact mapping and Inhouse file format. Some examples of specification documents that can be generated automatically from EC-Design are given in the screenshots below:

- The *Edifact UNSM mapping* documentation gives an overview of all information objects in a transaction that are mapped onto an Edifact UNSM (United Nations Standard Message). Edifact stands for Electronic Data Interchange For Administration, Commerce and Transport. An Edifact UNSM defines a preset structure for the exchange of information and is part of an Edifact TDID (Trade Data Interchange Directory). Each TDID consists of a message directory, segment directory, (composite) data element directory, and code lists.
  A mapping of a transaction onto an Edifact UNSM describes the conversion of information from a transaction to the syntax and structure of an Edifact UNSM. A transaction can be mapped to several different Edifact UNSMs.

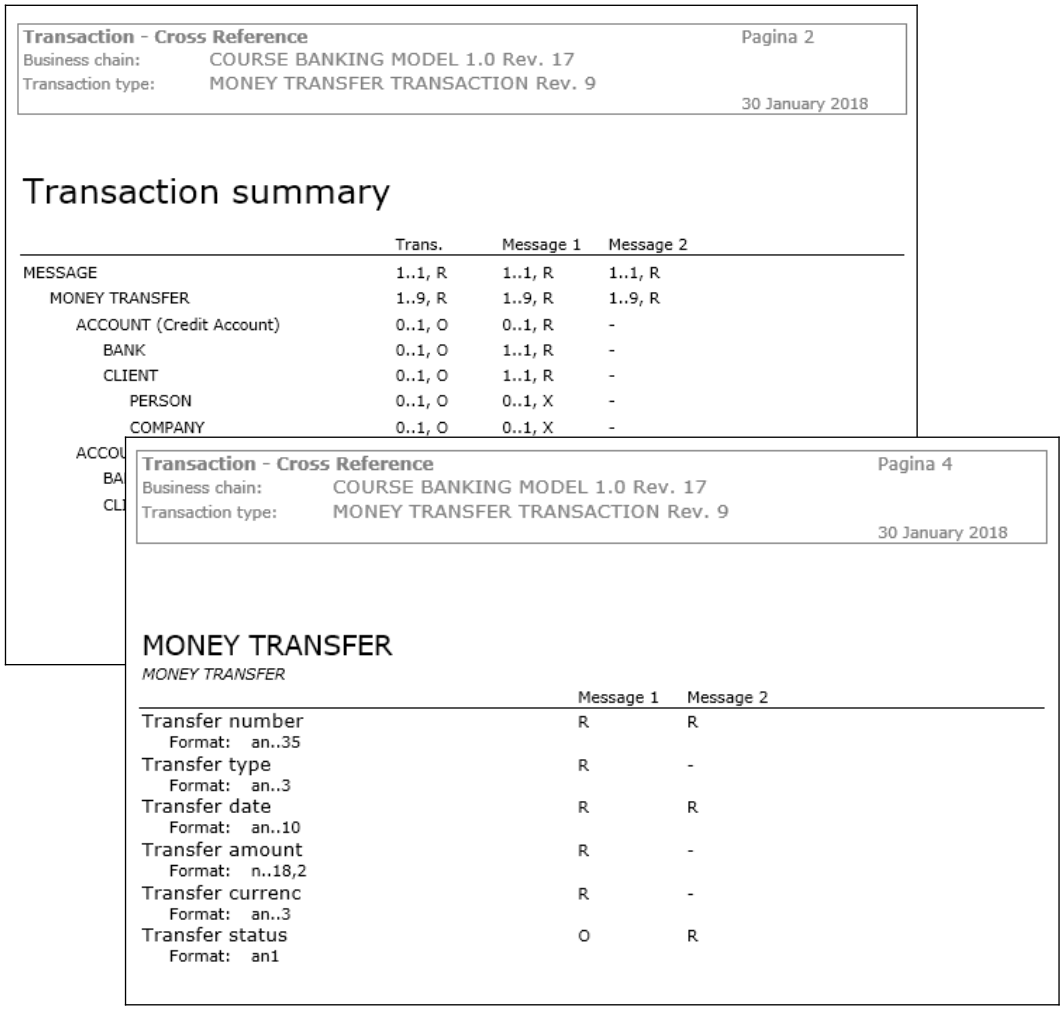


Transaction: Hierarchic — Page 9
Business chain: COURSE BANKING MODEL 1.0
Transaction type: MONEY TRANSFER TRANSACTION
Edifact directory: D96A — 9 March 2006

UNB UNH RFF ... UNT UNZ

RFF(TF).**FII[6]**, FINANCIAL INSTITUTION INFORMATION, C, 2x				a.money transfer - account (debit account) - bank, O, 1x b.money transfer - account (debit account), O, 1x
3035	PARTY QUALIFIER	M	an..3	*'DEB' (user defined)*
C078	ACCOUNT IDENTIFICATION	C		
3194	Account holder number	C	an..35	*b.Account number (R, an..35)*
3192	Account holder name	C	an..35	-
3192	Account holder name	C	an..35	-
6345	Currency, coded	C	an..3	-
C088	INSTITUTION IDENTIFICATION	C		
3433	Institution name identification	C	an..11	*a.Swift code (R, an..11)*
1131	Code list qualifier	C	an..3	-
3055	Code list responsible agency, coded	C	an..3	-
3434	Institution branch number	C	an..17	-
1131	Code list qualifier	C	an..3	-
3055	Code list responsible agency, coded	C	an..3	-
3432	Institution name	C	an..70	*a.Bank name (O, an..35)*
3436	Institution branch place	C	an..70	*a.Bank branch (O, an..35)*
3207	COUNTRY, CODED	C	an..3	-

- The *transaction cross-reference* documentation gives an overview of how a transaction is used in different ways by its related functional messages.



Transaction - Cross Reference — Pagina 2
Business chain: COURSE BANKING MODEL 1.0 Rev. 17
Transaction type: MONEY TRANSFER TRANSACTION Rev. 9
30 January 2018

Transaction summary

	Trans.	Message 1	Message 2
MESSAGE	1..1, R	1..1, R	1..1, R
MONEY TRANSFER	1..9, R	1..9, R	1..9, R
ACCOUNT (Credit Account)	0..1, O	0..1, R	-
BANK	0..1, O	1..1, R	-
CLIENT	0..1, O	1..1, R	-
PERSON	0..1, O	0..1, X	-
COMPANY	0..1, O	0..1, X	-

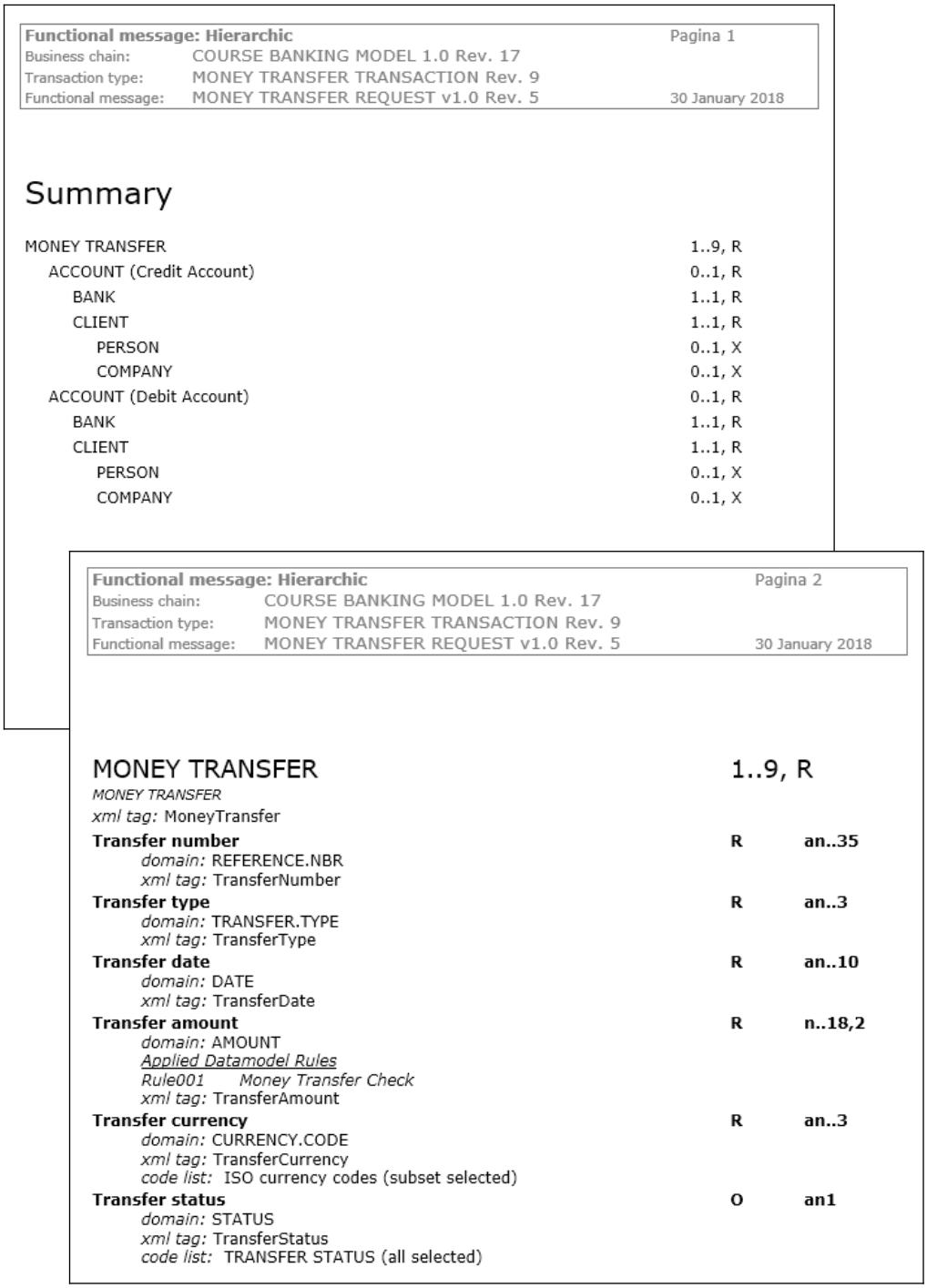
Transaction - Cross Reference — Pagina 4
Business chain: COURSE BANKING MODEL 1.0 Rev. 17
Transaction type: MONEY TRANSFER TRANSACTION Rev. 9
30 January 2018

MONEY TRANSFER

MONEY TRANSFER

	Message 1	Message 2
Transfer number Format: an..35	R	R
Transfer type Format: an..3	R	-
Transfer date Format: an..10	R	R
Transfer amount Format: n..18,2	R	-
Transfer currenc Format: an..3	R	-
Transfer status Format: an1	O	R

- The *functional message hierarchic* documentation gives a detailed specification of a functional message. All aspects modeled for the functional message are part of the documentation, such as data type formats, code list subsets, attribute statuses, repeat count for hierarchical relationships, relevant business rules, etc.



Functional message: Hierarchic Pagina 1
Business chain: COURSE BANKING MODEL 1.0 Rev. 17
Transaction type: MONEY TRANSFER TRANSACTION Rev. 9
Functional message: MONEY TRANSFER REQUEST v1.0 Rev. 5 30 January 2018

Summary

MONEY TRANSFER	1..9, R
ACCOUNT (Credit Account)	0..1, R
BANK	1..1, R
CLIENT	1..1, R
PERSON	0..1, X
COMPANY	0..1, X
ACCOUNT (Debit Account)	0..1, R
BANK	1..1, R
CLIENT	1..1, R
PERSON	0..1, X
COMPANY	0..1, X

Functional message: Hierarchic Pagina 2
Business chain: COURSE BANKING MODEL 1.0 Rev. 17
Transaction type: MONEY TRANSFER TRANSACTION Rev. 9
Functional message: MONEY TRANSFER REQUEST v1.0 Rev. 5 30 January 2018

MONEY TRANSFER 1..9, R

MONEY TRANSFER
xml tag: MoneyTransfer

Transfer number **R** **an..35**
domain: REFERENCE.NBR
xml tag: TransferNumber

Transfer type **R** **an..3**
domain: TRANSFER.TYPE
xml tag: TransferType

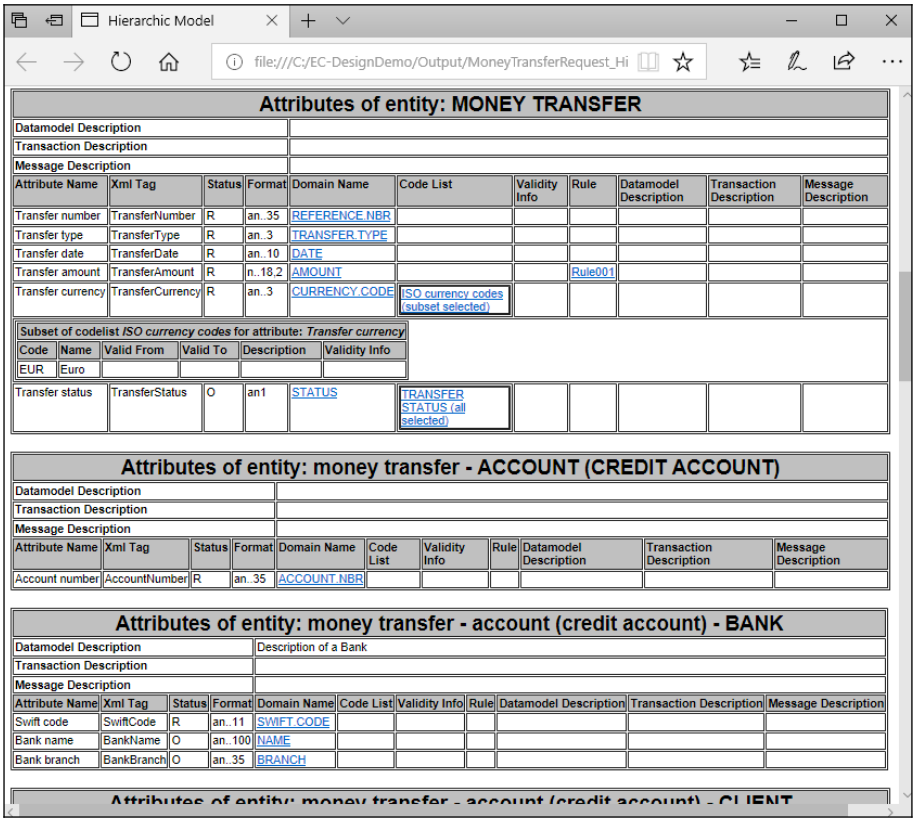
Transfer date **R** **an..10**
domain: DATE
xml tag: TransferDate

Transfer amount **R** **n..18,2**
domain: AMOUNT
Applied Datamodel Rules
Rule001 Money Transfer Check
xml tag: TransferAmount

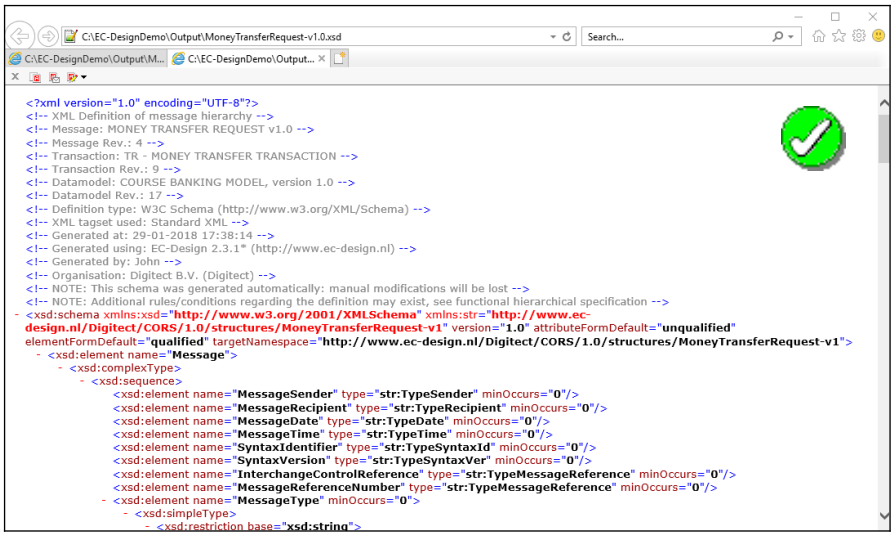
Transfer currency **R** **an..3**
domain: CURRENCY.CODE
xml tag: TransferCurrency
code list: ISO currency codes (subset selected)

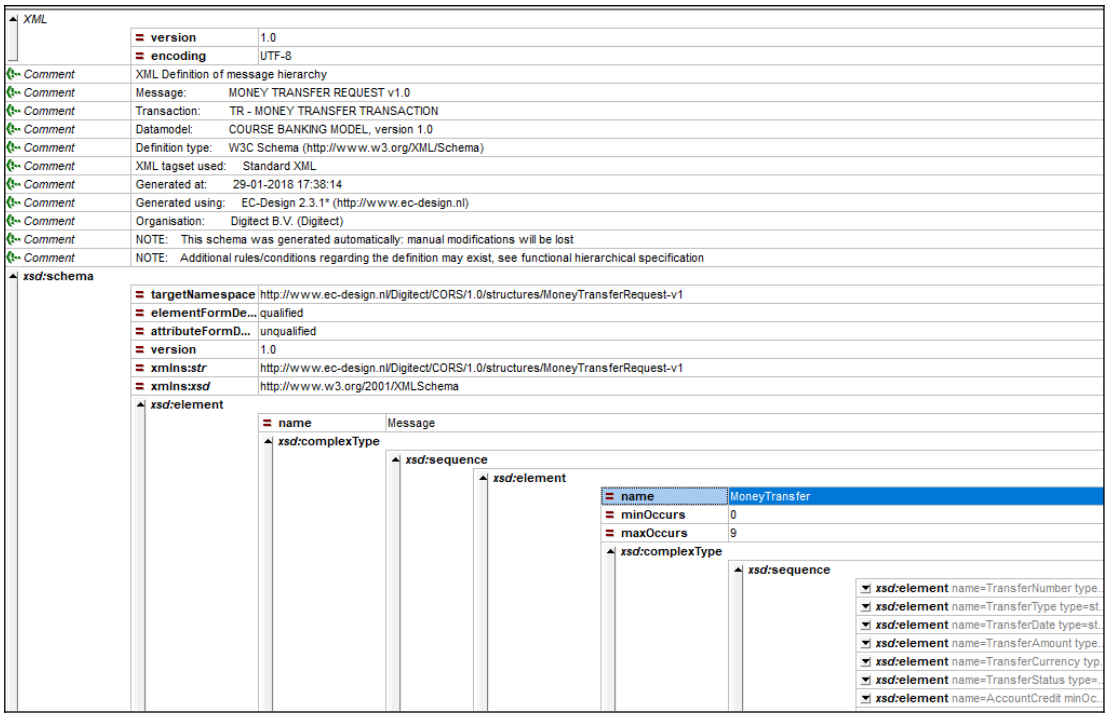
Transfer status **O** **an1**
domain: STATUS
xml tag: TransferStatus
code list: TRANSFER STATUS (all selected)

- The *functional message HTML* documentation is very similar to the *functional message hierarchic* documentation, only then its output format is readable for standard Internet web browsers.



- The *functional message XML Schema* technical documentation is an exact specification of the functional message in terms of the international W3C XML Schema standard. This output format is intended for machines more so than for humans to read. The functional message XML Schema can be fed straight into any conventional XML, XML Schema parser, such that the operational environment is configured automatically with the functional message modeled in EC-Design.




XML
version 1.0
encoding UTF-8
Comment XML Definition of message hierarchy
Comment Message: MONEY TRANSFER REQUEST v1.0
Comment Transaction: TR - MONEY TRANSFER TRANSACTION
Comment Datamodel: COURSE BANKING MODEL, version 1.0
Comment Definition type: W3C Schema (http://www.w3.org/XML/Schema)
Comment XML tagset used: Standard XML
Comment Generated at: 29-01-2018 17:38:14
Comment Generated using: EC-Design 2.3.1* (http://www.ec-design.nl)
Comment Organisation: Digitect B.V. (Digitect)
Comment NOTE: This schema was generated automatically: manual modifications will be lost
Comment NOTE: Additional rules/conditions regarding the definition may exist, see functional hierarchical specification
xsd:schema
targetNamespace http://www.ec-design.nl/Digitect/CORS/1.0/structures/MoneyTransferRequest-v1
elementFormDe... qualified
attributeFormD... unqualified
version 1.0
xmlns:str http://www.ec-design.nl/Digitect/CORS/1.0/structures/MoneyTransferRequest-v1
xmlns:xsd http://www.w3.org/2001/XMLSchema
xsd:element
name Message
xsd:complexType
xsd:sequence
xsd:element
name MoneyTransfer
minOccurs 0
maxOccurs 9
xsd:complexType
xsd:sequence
xsd:element name=TransferNumber type..
xsd:element name=TransferType type=st..
xsd:element name=TransferDate type=st..
xsd:element name=TransferAmount type..
xsd:element name=TransferCurrency typ..
xsd:element name=TransferStatus type=..
xsd:element name=AccountCredit minOc..

EC-Design benefits

- *Design and specify comprehensive, interoperable and certifiable information exchanges among independent information systems.*
- *Design and specify information exchanges based on international modeling standards:*
  UN / CEFACT Modeling Methodology (UMM)
  Unified Modeling Language (UML)
  ISO 11179, ISO 17113
- *Automatically generate multiple exchange syntaxes:*
  XML, XML Schema, JSON, JSON Schema, Edifact, Inhouse file format.
- *Specify any number of information exchanges (messages or web services) in a 100% consistent manner, based on one shared enterprise/business data model.*
- *Save time and money during analysis and implementation of information exchanges, and at the same time, greatly enhance the quality and ease the strain of change management.*

Contact us

If you have any questions regarding EC-Design, or when you are interested in purchasing a single or multi-user license, please do not hesitate to contact us.

May you enjoy working with EC-Design as much as we do.

Digitect B.V.
The Netherlands

mail@digitect.nl
www.digitect.nl
www.ec-design.nl