190P

C-Design

shi SUWL 1, 1 x, R Koke SUWL 1, 1 x, R dvet, 1, 1 x, R

Adves butenland
 Adves butenland
 Adves butenland

elefoonwammer Client, 0, 2 x, 0 mail adies client, 0, 1 x, 0 atonalitet, 1, 1 x, R

Parti SUWI, 1, 1 x, R

ng albeidinpasong, 1, * x, l Itol en realiteition, 1, 1 x, R

op de arbeidiplaats 1 1 v R

ericepinaati erit, 1.1 x, 8

EC-Design

The Enterprise Application Integration Standard

Statun Athibute No

♥ R
 ♥ R
 ♥ R
 ♥ 0

HE SUWIC

Entries and altributes

Adres buterland Advestment inte

Artwoordnummeradie: Arbeidugeschiktheid Clie Arbeidungst 6 walkoste Arbeidungst 6 walkoste Arbeidungst tonentabe Arbeidungst onentabe

]Arbeidtverhouding]Bedrijfsvereniging

Beendiging ogen bedrif Beendiging huwelijk/gen To ut

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:

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.

🔀 Datamodel - COURSE BAN	KING MODEL	v1.0 - Rev. 17* [Eng	lish]									, Β Σ	3	
Entities and attributes		Relations	Relations		Domains		Codelists		Diagram		R	Rules		
Name	Standard XM	L Code	Tag	Description	N	Name Stan		Standa	Standard XML		Domain name	Codelist]	
ACCOUNT	Account	ACCO	ACCNT		4	Transfer num	er	TransferNumber		an35	REFERENCE.N	l none		
BANK	Bank	BANK	BANK	Description of a Bar	4	Transfer type		Transfe	аТуре	an3	TRANSFER.TY	none		
CLIENT	Client	CLIENT	CLNT		4	Transfer date		Transfe	erD ate	an10	DATE	none		
COMPANY	Company				🔷 Transfer amount		Transfe	erAmount	n18,2	AMOUNT	none			
MESSAGE	Message		MSG		Transfer currency		Transfe	erCurrency	an3	CURRENCY.CO) 1			
MONEY TRANSFER	MoneyTransi	er MON	MNY		Transfer status		Transfe	erStatus	an1	STATUS	1			
	reison				Datamodel - COURSE BANKING MODEL v1 Entities and attributes Rv Selected domain CURRENCY.CODE an. 3		MODEL v1.0 Rela) - Rev. 17 [Englis] tions	h] Domains	Codelists	Diagram	Rules	~	
						Codelist Name	Id	Externa	I Responsible Ag	ency Code V	alue Code Name	Dep	aiption	_
						< .			_	Series Contraction	British Pound US Dollar			>
<				>	<							>		

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.



• 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.

ខែ Functional Message - MONEY TRANSFER REQUEST v1.0 (CORS 1.0) TR) - Rev	/. 4 [Englisl	h]					- • ×	1
HESSAGE, 11 x, R	Ge	neral	Entity	Attri	ibutes	М	apping	Rules	
	Status ◆ R ◆ R ◆ R ◆ R ◆ R ◆ R ◆ R	Attribute N Transfer n Transfer ty Transfer d Transfer a Transfer s Transfer st	ame umber pe ate mount urrency tatus	Standard XML TransferNumber TransferType TransferDate TransferAmount TransferCurrenc TransferCurrenc	r a a : n xy a a	ata Type n. 35 n. 3 n. 10 18,2 n. 3 n1	Domain Name REFERENCE.N TRANSFER.TY DATE AMOUNT CURRENCY.CO STATUS	Codelist V none none none 1/1 A 1/1 A	
COMPANY, 0.1 x.X			Codelist Subset Name ISO currency cod	Code les	External F N	Value Gal CNY CNY CNY EUR Gal BP Cal GBP	Name Chinese Yuar Euro British Pound US Dollar	De	scription
			Select All	Deselect All] [OK	Cancel
	۲							>	

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, 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.

Transactio	on: Hierarchie			Page 9					
Business ch	ain CC	URSE BANKI	NG MO	ODEL 1.0					
Transaction	tune: Mi	ONEV TDANG	FFD TI	DANGACTION					
Transaction	type. IVIC	JINET IKAINS	PER II	MINDACTION					
Edifact directory: D96A 9 March 2									
1									
RFF(17).FII[6], FINANCIAL INSTITUTION INFORMATION, C, 2 × a money transfer - account (debit account) - <u>bank</u> , O, 1×									
				to many any terms after a second (debit second) at					
				b.money transfer - <u>account</u> (debit <u>account</u>), 0, 1x					
3035 par	TY QUALIFIER	М	an3	b.money transfer - <u>account</u> (debit <u>account</u>), 0, 1x [DEB' (user defined)					
3035 PAR	TY QUALIFIER	м	an3	b.money transfer - <u>account</u> (debit <u>account</u>), <u>0</u> , <u>1x</u> <i>'DBB' (user defined)</i>					
3035 PAR C078 ACO 3194 Ac	TY QUALIFIER OUNT IDENTIFICATION	M	an.3	b.money transfer - <u>account</u> (debit <u>account</u>), $\overline{0}$, $\overline{1x}$ DBB' (user defined) b Account number (R. an. 35)					
3035 PAR C078 ACO 3194 Ac 3192 Ac	TY QUALIFIER OUNT IDENTIFICATION count holder number	M C C C	an3 an35 an35	b.money transfer - <u>account</u> (debit <u>account</u>), 0, 1x "DBB" (user defined) b.Account number (R, an. 35)					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac	TY QUALIFIER OUNT IDENTIFICATION count holder number count holder name count holder name	M C C C C	an3 an35 an35 an.35	b.money transfer - <u>account</u> (debit <u>account</u>), <u>0</u> , <u>1x</u> <i>UBB'</i> (user defined) b.Account number (R, an35) -					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 3192 Ac 3192 Ac 6345 Cus	TY QUALIFIER OUNT IDENTIFICATION count holder number count holder name count holder name rency, coded	M C C C C C C	an3 an35 an35 an35 an3	b.money transfer - <u>account</u> (debit <u>account</u>), 0, 1x "DBB" (user defined) b.Account number (R, an35) - -					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 3192 Ac 6345 Cus C022 WCT	TY QUALIFIER OUNT IDENTIFICATION count holder number count holder name count holder name rency, coded	M C C C C C C C	an3 an35 an35 an35 an3	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0</u> , 1x 'DBB' (user defined) b.Account number (R, an35) - -					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 6345 Cus C0888 INST 2423	TY QUALIFIER OUNT IDENTIFICATION count holder number count holder name count holder name rency, coded ITUTION IDENTIFICATION	M C C C C C C C C C C C C C C C C C C C	an3 an35 an35 an35 an3	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0</u> , <u>1x</u> <i>'UEB' (user defined)</i> b.Account number (R, an35) - - - - - - -					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 6345 Cus C088 INST 3433 Ins 3131 Cos	TY QUALIFIER DUNT IDENTIFICATION sount holder name count holder name count holder name rency, coded rency, coded truttion IDENTIFICATIO thiblion name identification do lot consider	M C C C C C C N C C	an3 an35 an35 an35 an3 an3	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0</u> , <u>1x</u> <i>'UBB'</i> (user defined) b.Account number (R, an35) - - a.Swift code (R, an11)					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 6345 Cu C088 INST 3433 Ins 1131 Co 3055 Co	TY QUALIFIER OUNT IDENTIFICATION count holder number count holder name rency, coded ITUTION IDENTIFICATION Ibution name identification de list qualifier de list qualifier	M C C C C C C C C C C C C C C C C C C C	an3 an35 an35 an35 an3 an11 an3	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0, 1x</u> 'DBB' (user defined) b.Account number (R, an35) - - - a.Swift code (R, an11)					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 6345 Cui C088 INST 3433 Ins 1131 Coi 3055 Coi	TY QUALIFIER OUNT IDENTIFICATION count holder name count holder name count holder name tenny, coded ITUTION IDENTIFICATION thitticon name identification de list qualifier de list qualifier de list qualifier de list qualifier thitticon benchourbert	M C C C C C N N C C C C C C C C C C C C	an.35 an.35 an.35 an.3 an.3 an.3 an.3 an.3	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0</u> , <u>1x</u> <i>'UHB'</i> (user defined) b.Account number (R, an35) - - a.Swift code (R, an11) -					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 3192 Ac 6345 Cus C088 INST 3433 Ins 3433 Ins 3433 Ins 3434 Ins 3434 Ins	TY QUALIFIER DUNT IDENTIFICATION count holder number count holder name rency, coded ITUTION IDENTIFICATIO ibution name identification de list qualifier de list responsible agency, tibution branch number de list responsible	M C C C C C C C C C C C C C C C C C C C	an.3 an.35 an.35 an.3 an.3 an.3 an.3 an.3 an.3 an.3	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0, 1x</u> 'DBB' (user defined) b.Account number (R, an35) - - a.Swift code (R, an11) -					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 3192 Ac 3192 Ac 3193 Ins 1131 Co 3055 Cou 3434 Ins 1131 Co 3055 Cou	TY QUALIFIER OUNT IDENTIFICATION count holder name count holder name teneng, coded ITUTION IDENTIFICATION thistion name identification de list qualifier de list qualifier de list qualifier de list qualifier de list qualifier de list qualifier de list qualifier	M C C C C C C C C C C C C C C C C C C C	an3 an35 an35 an3 an3 an3 an3 an3 an17 an3 an17 an3	b.monéy transfer - <u>account</u> (debit <u>account</u>), 0, 1x DEB' (user defined) b.Account number (R, an35) - - a.Swift code (R, an11) - -					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 3192 Ac 6345 Cu C088 INST 3433 Ins 1131 Co 3055 Co 3434 Ins 1131 Co 3055 Co 3434 Ins 1131 Co 3055 Co 3434 Ins 1131 Co	TY QUALIFIER DUNT IDENTIFICATION count holder number count holder name count holder name tency, coded TIUTION IDENTIFICATION tibution name identification de list qualifier de list qualifier	M C C C C C C C C C c C C C C C C C C C	an3 an35 an35 an3 an3 an3 an3 an17 an3 an3 an3 an3	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0, 1x</u> <i>'UBB'</i> (user defined) b.Account number (R, an35) - - a.Swift code (R, an11) - - - - - - - - - - - - -					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 6345 Cui C088 INST 3433 Ins 1131 Coi 3055 Coi 3434 Ins 3055 Coi 3432 Ins 3436 Ins	TY QUALIFIER DUNT IDENTIFICATION count holder number count holder name count holder name rency, coded ITUTION IDENTIFICATION ibution name identification de list qualifier de list responsible agency, tibution branch number le list qualifier de list responsible agency, tibution name tibution name bition panch place	M C C C C C C C C C C C C C C C C C C C	an3 an35 an35 an3 an3 an3 an11 an3 an17 an3 an7 an3 an7	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0, 1x</u> 'DBB' (user defined) b.Account number (R, an35) - - a.Swift code (R, an11) - - - a.Bank name (O, an35) a. Bank name (O, an35)					
3035 PAR C078 ACC 3194 Ac 3192 Ac 3192 Ac 6345 Cuu C088 INST 1131 Co 3055 Co 3432 Ins 3436 Ins	TY QUALIFIER OUNT IDENTIFICATION count holder name count holder name count holder name tenny, coded TTUTION IDENTIFICATION thution name identification de list qualifier de lis	M C C C C C n coded C C C C C C C C C C C C C C C C C C	an3 an35 an35 an3 an3 an11 an3 an17 an3 an70 an70	b.monéy transfer - <u>account</u> (debit <u>account</u>), <u>0</u> , <u>1x</u> <i>DBB'</i> (user defined) b.Account number (R, an35) - - a.Swift code (R, an11) - - - - - - - - - - - - -					

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

Fransaction	1 - Cross Reference				Pagina 2	
usiness chair	COURSE B	ANKING MODEL 1	.0 Rev. 17		r uginu z	
ransaction tv	vpe: MONEY TR	ANSFER TRANSAC	TION Rev. 9			
,					30 January	2018
Transa	action sum	mary				
		Trans.	Message 1	Message	2	
1ESSAGE		11, R	11, R	11, R		
MONEY TR	ANSFER.	19, R	19, R	19, R		
ACCOU	INT (Credit Account)	01, O	01, R	-		
BAN	NK	01, O	11, R	-		
CLI	ENT	01, O	11, R	-		
	PERSON	01, O	01, X	-		
-	COMPANY	01, O	01, X	-		
	MONEY TRAI	NSFER				30 January 2018
	MONEY TRANSFER		м	essage 1	Message 2	
	Transfer number Format: an35		R	-	R	
	Transfer type Format: an3		R		-	
	Transfer date Format: an10		R		R	
			R		-	
	Format: n18,2					
	Format: n18,2 Transfer currenc Format: an3		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: C	URSE BANKING MODEL 1.0 Rev. 17		
Functional message: M	ONEY TRANSFER TRANSACTION REV. 9	30 January 2018	2
runcuonarmessage. m		50 January 2010	<u>, </u>
Summary			
MONEY TRANSFER		19, R	
ACCOUNT (Credit Ac	count)	01, R	
BANK		11, R	
CLIENT		11, R	
PERSON		01, X	
COMPANY		01, X	
ACCOUNT (Debit Acc	ount)	01, R	
BANK		11, R	
CLIENT		11, R	
PERSON		01, X	
COMPANY		01, X	
Eunctional m	accado: Hiorarchic	Dag	ina 2
Business chain:	COURSE BANKING MODEL 1.0 Rev. 17	Fay	1110 2
Transaction type	: MONEY TRANSFER TRANSACTION Rev. 9		
Functional mess	age: MONEY TRANSFER REQUEST v1.0 Rev. 5	30 3	January 2018
MONEY T MONEY TRANSFU xml tag: Mone	RANSFER ^{;R} yTransfer	19	, R
Transfer num		R	an35
xml tag:	TransferNumber		
Transfer type		R	an3
domain:	TRANSFER.TYPE TransferType		
Transfer date		R	an10
domain:	DATE		
xml tag:	TransferDate		10.5
domain:	amount	к	n18,2
Applied Rule001	<u>Datamodel Rules</u> Money Transfer Check		
xml tag:	TransferAmount	р	
domain:	CURRENCY.CODE	ĸ	an <i>3</i>
xml tag:	TransferCurrency		
code list	ISO currency codes (subset selected)	0	an1
domain:	STATUS	0	a111
xml tag: code list	TransferStatus : 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.

6 8 8	Hierarchie	: Mod	el	×	+	\sim							-	
$\leftarrow \ \rightarrow$	Ŭ	ት	0	file;,	///C:/EC-I	DesignDe	mo/Outpi	it/Mone	eyTransfer	Request_	ні 🛄 🕁	∑_=	h	È
Attributes of entity: MONEY TRANSFER														
Datamodel Desc	ription													
Transaction Des	cription													
Message Descri	ption													
Attribute Name	Xml Tag		Status	Forma	at Domain	Name	Code List		Validity Info	Rule	Datamodel Description	Transaction Description	i Me De	essage scription
Transfer number	TransferNu	imber	R	an35	REFERE	NCE.NBR								
Transfer type	TransferTy	ре	R	an3	TRANSF	ER.TYPE	ļ			<u> </u>				
Transfer date	TransferDa	ite	R	an10	DATE		<u> </u>							
Transfer amount	TransferAn	nount	R	n18,2	2 AMOUN	<u> </u>	<u> </u>			Rule001		ļ		
Transfer currency	TransferCu	irrency	R	an3	CURREI	VCY.CODE	ISO curre (subset se	ncy codes elected)	2					
Subset of codel	list ISO curr	ency c	odes fo	r attrik	oute: Trans	fer curren	¢y							
Code Name	Valid From	Valie	d To 🛛	Descri	ption Va	ilidity Info								
EUR Euro														
Transfer status	TransferSt	atus	0	an1	STATUS		TRANSFE	R						
							STATUS selected)	all						
							<u>ن</u>							
	Attril	bute	s of	ent	ity: m	oney t	ransfe	r - AC	COUN	IT (CF	REDIT AC	COUNT)	
Datamodel Desc	ription													
Transaction Des	cription													
Message Descri	ption													
Attribute Name	Xml Tag	St	atus Fo	ormat	Domain Na	ime Code List	e Valid Info	ity R	ule Datamo Descrip	odel otion	Transaction Description	on on	Messag Descrip	e tion
Account number	AccountNun	nber R	an	n35	ACCOUNT	.NBR								
	Attrib	utes	s of (enti	tv: mo	nev tr	ansfer	- acc	count (credit	account) - BANK	<	
Datamodel Desc	ription			Des	cription of a	Bank						/ _/	-	
Transaction Des	cription				support of e									
Message Descri	ption													
Attribute Name	Xml Tag	Status	Forma	at Don	nain Name	Code List	Validity In	fo Rule	Datamodel	Descriptio	n Transaction	Description N	lessage	Description
Swift code	SwiftCode	R	an11	SW	FT.CODE									
Bank name	BankName	0	an10	0 NAM	ЛЕ			j – j						
Bank branch	BankBranch	0	an35	BRA	ANCH			Ť						
	Attrib	itae	ofo	ntit		nov tre	nefor	- 200	ount la	prodit	account)		т	

 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.

ChEC_DesignDamo\Output\MoneyTransferRequest.v10vrd	- 6	Search	- □ ×
		Jearchin	
<pre><?xml version="1.0" encoding="UTF-8"?> <!-- XML Definition of message hierarchy ---> <!-- Message: MONEY TRANSFER REQUEST V1.0--> <!-- Transaction: TR - MONEY TRANSFER TRANSACTION--> <!-- Transaction: TR - MONEY TRANSFER TRANSACTION--> <!-- Transaction: TR - MONEY TRANSFER TRANSACTION--> <!-- Datamodel: COURSE BANKING MODEL, version 1.0--> <!-- TAML tagget used: Standard XML--> <!-- Constructed th: 20-01210 17:28:14--> <!--</td--><td></td><td></td><td>Ø</td></pre>			Ø
CI- Generated using: EC-Design 2.3.1* (http://www.ec-design.nl)> <i (http:="" 2.3.1*="" ec-design="" generated="" using:="" www.ec-design.nl)=""> <i (digitect)="" d.v.="" digitect="" organisation:=""> <i automatically:="" generated="" manual="" modifications="" note:="" schema="" this="" w<br="" was=""><i additional="" conditions="" definition="" exist,="" f<br="" may="" note:="" regarding="" rules="" see="" the="">< cxsdischema xmins:xsd="http://www.w3.org/2001/XMLSchema" xmins design.nl/Digitect/CORS/1.0/structures/MoneyTransferRequeet+v1" v elementFormDefault="qualified" targetNamespace="http://www.ec-design < cxsd:element name="http://wespace">http://www.ec-design</i></i></i></i>	Il be lost> inctional hierarchica str="http://www. rsion="1.0" attribut .nl/Digitect/CORS	I specification> ec- teFormDefault="unqualif /1.0/structures/Mone	fied" eyTransferRequest-v1">
<pre>- <xsd:complextype> - <xsd:sequence></xsd:sequence></xsd:complextype></pre>	er" minOccurs="0"/	>	
<pre><xsd:element <br="" messagenect"="" name="MessageNeccipient" type="str:TypeInter"><xsd:element <br="" name="MessageTime" type="str:TypeTime"><xsd:element interchangecontrolreference"="" name="SyntaxVectifient" syntaxvection"="" type="<br"><xsd:element messagereferencenumber"="" name="MessageReferenceNumber" str<br="" type="str
<xsd:element name="><xsd:element name="str
<xsd:element name=" str<br=""><xsd:element minoccurs="<br" name="str
<xsd:element name</td><td>:pent">inOccurs="0"/> inOccurs="0"/> axId" minOccurs="0 :Ver" minOccurs="0 'str:TypeMessageF :TypeMessageRefe</xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></xsd:element></pre>	"/> "/> Reference" minOccurs=" erence" minOccurs="0"/	" 0 "/> '>	
- <xst:sement minoccurs="0" name="Messagerype"> - <xst:simpletype> -</xst:simpletype></xst:sement>			

▲ 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	Datamodet 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://w	ww.ec-design.nl)							
Comment	Organisation: Digite	ect B.V. (Digitect)								
(Comment	NOTE: This schema was generated automatically: manual modifications will be lost									
Comment	NOTE: Additional rule:	s/conditions regarding th	ne definition may e	xist, see functional h	erarchical specifica	tion				
 xsd:schema 										
	targetNamespace	http://www.ec-design.	nl/Digitect/CORS/1	.0/structures/Money	ransferRequest-v1					
	elementFormDe	qualified								
	attributeFormD	unqualified								
	version	1.0								
	= xmins:str	http://www.ec-design.	nl/Digitect/CORS/1	.0/structures/Money	ransferRequest-v1					
	= xmins:xsd	http://www.w3.org/20	01/XMLSchema							
	xsd:element									
		= name	Message							
		xsd:complexType								
			xsd:sequer	ce						
				xsd:element						
					= name	MoneyTransfer				
					= minOccurs	0				
					= maxOccurs	9				
					 xsd:complet 	ехТуре				
						xsd:sequence				
							xsd:element name=TransferNumber type.			
							xsd:element name=TransferDate type=st.			
							xsd:element name=TransferStatus type=.			
							✓ xsd:element name=AccountCredit minOc.			
		Contraction of the second s	1.000		a second s					

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, 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. P.O. box 30 1540 AA Koog aan de Zaan The Netherlands

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