



DPC: 24/30487286 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: 16 December 2024

Origin: International

Latest date for receipt of comments: 08 February 2025

Project No. 2024/00395

Responsible committee: B/555 Construction design, modelling and data exchange

Interested committees:

Title: Draft BS EN ISO 29481-2 Building information models - Information delivery manual. Part 2: Interaction framework.

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 08 February 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 29481-2

Building information models — Information delivery manual —

Part 2: Interaction framework

*Modèles des informations de la construction — Protocole
d'échange d'informations —*

Partie 2: Cadre d'interaction

ICS: 35.240.67

ISO/TC 59/SC 13

Secretariat: **SN**

Voting begins on:
2024-12-16

Voting terminates on:
2025-03-10

This document is circulated as received from the committee secretariat.

ISO/CEN PARALLEL PROCESSING

Reference number
ISO/DIS 29481-2:2024(en)

Full document details: 24/30487286 DC

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.

© ISO 2024



Documents

Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO 2024

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	v
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
3.1 Terms defined in ISO 29481-1	1
3.2 Other terms and definitions	3
4 Interaction Framework	3
4.1 General	3
4.2 Information Management and the interaction framework	4
4.3 Purpose of an Interaction Framework	5
4.4 Hierarchical structure of an interaction framework	6
4.4.1 Roles	6
4.4.2 Transactions	7
4.4.3 Messages	8
4.4.4 Complex and Simple Elements	9
4.5 Establishing a Digital IDM communication Project	10
4.5.1 Modelling the interaction framework	10
4.5.2 Creating and using the interaction framework schema	11
4.5.3 Using a project specific message to link people to roles	11
4.5.4 Executing digital IDM communication	12
4.5.5 Validating digital IDM communication	13
4.5.6 Signing messages with advanced electronic signatures	13
4.5.7 Changing the project during execution	14
4.5.8 Archiving digital IDM communication	14
5 Schemas for validating interaction frameworks and messages	15
5.1 Introduction	15
5.1.1 Basic principles	16
5.1.2 Element types in the interaction framework schema and _2 EXPRESS source file	16
5.1.3 Introduction	16
5.1.4 Primary element types	18
5.1.5 Secondary element types	27
5.1.6 References	30
5.2 Element types in the interaction message schema and the _5 EXPRESS source file	37
5.2.1 Introduction	37
5.2.2 Primary element types	38
5.2.3 Secondary element types	43
5.2.4 References	45
Annex A (normative) Interaction framework schema definition	52
Annex B (normative) Templates definition	53
Annex C (informative) Promotor algorithm	54
Annex D (informative) Example interaction framework XML of a simplified design office use case	55
Annex E (informative) Example project specific message XML of a simplified design office use case	72
Annex F (informative) Example message XML of a simplified design office use case	79
Bibliography	90

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO *had not* received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents. ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 59, *Buildings and civil engineering works*, Subcommittee SC 13, *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM)*.

ISO 29481 consists of the following parts, under the general title *Building information models — Information delivery manual*:

- *Part 1: Methodology and format*
- *Part 2: Interaction framework*
- *Part 3: Data Schema*

This second edition cancels and replaces the first edition (ISO 29481-2:2012) and introduces updates that better integrate the interaction framework within the concept of digital IDM communication. It also aligns terminology and practices with the latest in the IDM series and other related standards

Introduction

Collaboration between the participants involved during the lifecycle of built assets is pivotal to the efficient delivery and operation of assets. Organizations collaborate within specific use cases to achieve higher levels of quality and greater re-use of existing knowledge, information experience and resources. A significant condition of collaboration is the opportunity to communicate, re-use and share information efficiently, and to reduce the risk of loss, contradiction or misinterpretation.

The information delivery manual (IDM) series provides significant assistance in making the most of information management. If the necessary information is available at the right time, the quality of the information is satisfactory and the right person is involved at the right time, the collaboration and outcome itself is greatly improved. Management and coordination of the parties involved depend on communication, therefore it shall be well structured, unambiguous, explicit and prompt. For this to happen, there must be a common understanding of the purpose, the processes, the actors involved and the required information.

This document focuses on the foundations for and execution of digital communication in accordance with the processes and information requirements of a use case. With a sharp focus on communication, this part of ISO 29481 offers a natural complement to standards that focus on information management such as ISO 19650, information containers such as ISO 21597 and information modeling, such as ISO 16739 and ISO 10303-239.

This document describes how to use various components of an Information Delivery Manual (IDM) for verifiable and traceable execution of digital communication. The resulting interaction framework enables standardization of digital communication in construction processes within any collaboration, both within and between organizations. It is applicable during the complete lifecycle of built assets and construction projects of all sizes and all levels of complexity. It provides a standard format that IT solutions may use. Supporting this standard in various ICT solutions means that various information management systems are connected. By doing so, it provides a basis for reliable information exchange and sharing for users, so that they can be confident that the information they send or receive is accurate and sufficient for the coordination activities they need to perform. This provides a basis for using Common Data Environment (CDE) solutions and workflows.

Building information models — Information delivery manual —

Part 2: Interaction framework

1 Scope

Document specifies a methodology for describing and managing interactions and a format for digital communication between actors in any use case associated with the management of an asset during all life cycle stages.

It therefore specifies

- a methodology that describes an interaction framework for a use case;
- an appropriate way to map responsibilities and interactions that provides a process context for information flow;
- a format in which the interaction framework should be specified and executed.

This document is intended to promote secure, verifiable, traceable and high-quality digital IDM communication between actors during all phases of the lifecycle of built assets, facilitate interoperability between software applications used, and to provide a basis for data- and process driven information exchange and traceability of communication.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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 29481-1, *Building information modelling — Information delivery manual — Part 1: Methodology and format*

ISO 29481-3, *Building information models — Information delivery manual — Part 3: Data schema*

ISO 10303-11, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*

ISO 40210, *Information technology — W3C SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)*

3 Terms and definitions

3.1 Terms defined in ISO 29481-1

For the purposes of this document, the following terms defined in ISO 29481-1 apply.

3.1.1

asset

item, thing or entity that has potential or actual value to an organization

[SOURCE: ISO 55000:2024, 3.1.1, modified - The notes to entry have been removed.]

3.1.2

use case

description of a process by defining a sequence of actions performed by one or more roles to realize a purpose

Note 1 to entry: a use case specifies the scope of the business context.

[SOURCE: ISO 24014-1:2021, 3.36, modified - the word "actor" has been replaced with "role". The words "and by the system itself" has been replaced with "to realize a purpose". The note to entry has been added.]

3.1.3

business context

environment in which actors undertake processes

Note 1 to entry: Such an environment includes all the roles, activities, and interactions, products and services under the actor's control.

Note 2 to entry: the business context always contains one or more business processes

3.1.4

business process

set of interrelated or interacting activities that use inputs to deliver an intended result

EXAMPLE Examples results include: agreements, systems, services, or products.

Note 1 to entry: a business process always contains one or more transactions.

[SOURCE: ISO 9000:2015, 3.4.1, modified - The notes to entry have been replaced with a note to entry. An example has been added.]

3.1.5

information delivery manual

IDM

specification of a use case using business context maps and exchange requirements

Note 1 to entry: Use case, business context maps, and exchange requirements may be collectively referred to as IDM components.

3.1.6

interaction map

business context map of the interactions related to a use case

3.1.7

information exchange

act of satisfying an information requirement or part thereof

[SOURCE: ISO 19650-1:2018, 3.3.7]

3.1.8

interaction

communicative action between two roles

3.1.9

transaction

sequence of interactions involving the exchange of information using messages

EXAMPLE processing of a request by one role on behalf of another

Note 1 to entry: a transaction is a message between the initiator and the executor.

3.1.10

role

functions being performed by an actor at a point in time

Note 1 to entry: The role of an actor is determined by action and outcome and not necessarily by the profession or trade followed by the actor.

3.1.11

message

information prepared for communication purposes

[SOURCE: ISO 5127:2017, 3.1.8.02]

3.2 Other terms and definitions

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

3.2.1

digital IDM communication

exchange of messages within the confines of a use case enabled by an interaction framework and project specific message

3.2.2

interaction framework

XML file for the elements of an interaction

Note 1 to entry: Elements of an interaction include roles, transactions, messages, and information units

3.2.3

interaction framework schema

XML schema definition file that the interaction framework shall comply with

3.2.4

project specific message

PSM

message placing the interaction framework in the context of a project and links people to organizations and roles

3.2.5

interaction message schema

XML schema definition file that the messages shall comply with

3.2.6

promotor

algorithm that generates an interaction framework schema from an interaction framework, with the support of a template definition used as input

3.2.7

templates definition

file containing a number of templates, independent of the specific interaction framework, for generating an interaction framework schema and an interaction message schema

4 Interaction Framework

4.1 General

This Clause explains the core concepts and principles for the development and use of an Interaction Framework for digital IDM communication. By following the methodology presented in this document users of software tools are able to model (in a machine interpretable XML file) the business context and exchange requirements of a specified use case and subsequently start to digitally communicate with each other based on that interaction framework.

4.2 Information Management and the interaction framework

Information management involves defining, producing, checking, and delivering information. Every team member in any business activity interacts with information management to some extent. The importance of documented responsibility and accountability in information management is crucial, because focusing on good information management leads to better project outcomes. To be effective, there shall be a shared understanding of the processes and the necessary information in each step for their execution.

An IDM captures the business context associated with a use case and defines a detailed specification of the information that a user fulfilling a particular role would need to provide at a particular stage within a business process in order to create a product or deliver a service.

Generally speaking, a use case involves delivering goods or providing services to people or institutions. These people and institutions are called the use case customers or clients. As soon as a customer or client requests the delivery of a batch of goods or the provision of a service, a chain of activities is set in motion to fulfil that request. Such a chain of activities is called a business context.

This leads to the basic framework of an IDM as illustrated in [Figure 1](#). Each use case has an associated business context and information requirements.

Information Delivery Manual

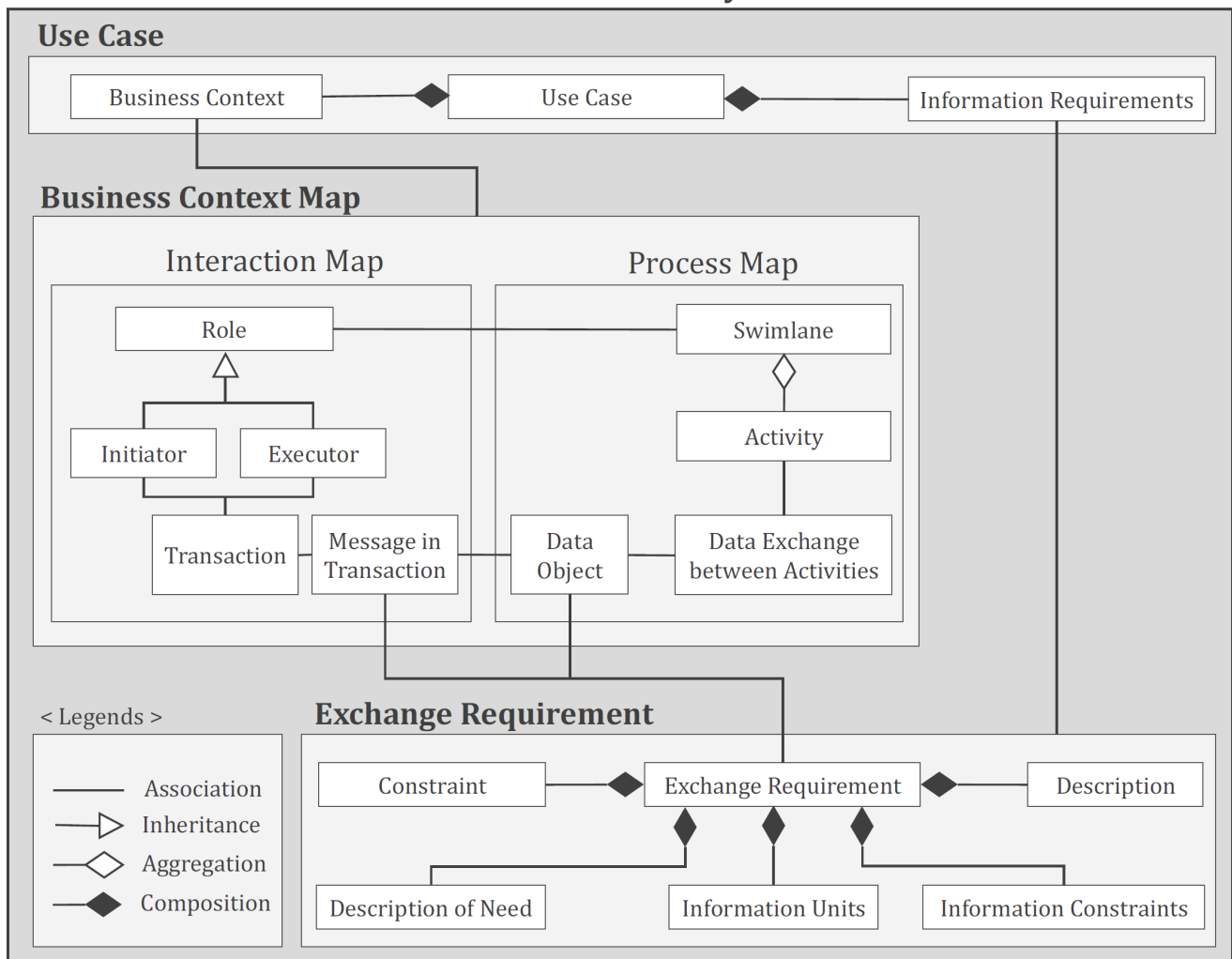


Figure 1 — IDM Basic framework

The business context is usually represented in an IDM by any combination of interaction maps and process maps, supported by textual representations. The analysis of the business context may be thought of as

a discovery process using those two mapping approaches since their purpose is to clearly show how the information exchange requirements for the use case have been derived.

An exchange requirement provides a description of the information to be exchanged between roles in plain language terms to support a particular business process in the relevant life cycle stage (within a business context). An exchange requirement may support either (or both) the communication of information essential to the design, construction, operation and deconstruction of an asset or it may support the communication of management information with messages that control project execution in relation to the life cycle management of the asset.

To deliver a service or produce a product within a use case, people perform various activities. Taken in their entirety, these form the business context. Often, within the business context there are chains of activities that provide sub products or services, these are called business processes. At certain points in time, information is needed by an actor to fulfil their task, or it is part of their responsibility to deliver specific information. When this information is transferred from one actor to another, an interaction occurs. The receiving actor determines if the information is complete and correct.

An information exchange therefore is never a single activity, it is a (digital) conversation between two actors in order to satisfy a specific information requirement.

This document focuses on formally structuring, recording and supporting these interactions and their requirements to achieve an outcome. This is known as performative communication.

4.3 Purpose of an Interaction Framework

The interaction framework is the core of digital IDM communication. It is a formal technical XML description that includes all elements necessary for interactions within use cases, such as roles, transactions, messages in transactions, and data elements in messages. The scope of an interaction framework aligns with the scope of a use case in an Information Delivery Manual (IDM).

Creating an interaction framework not only allows the identification of detailed information exchange requirements, but also facilitates (through compatible software) verifiable and traceable digital IDM communication between participants during all phases of the lifecycle of built assets. When properly set up, an interaction framework forms the foundation for software tools to enable users to communicate digitally and purposefully through messages based on the processes, roles, exchange requirements, and information defined by the IDM. Digital IDM communication through an interaction framework enables better process control, improved data management, increased transparency, enhanced verifiability of agreements, and a higher quality of the final product. A software tool utilizing an interaction framework structures, monitors, and stores communication by providing a shared digital communication archive. It clarifies responsibilities and decisions promptly, keeps project archives up-to-date and complete, facilitates faster and more flexible partnerships, and prevents miscommunication.

Since the process and content resides in the machine-readable interaction framework rather than being embedded in a software tool, a software solution implementing this method can support and improve collaboration of any use case without specialised software.

It also ensures interoperability between software tools that support this document, facilitating seamless messaging between applications. Every organization may freely choose their software tools that implements the interaction framework, ensuring secure messaging between them. This guarantees each party complete access to mutual communication through their own chosen software while shielding internal communication from the collaborating partners.

The basic principles, terms and concepts of the modelling of interactions that this document uses are based on DEMO¹⁾ (Design and Engineering Methodology for Organizations). DEMO is a method for analysing

1) DEMO originated from science (Maastricht University, Delft University of Technology) and builds on various theories including systems theory (Simon, 1962), philosophy of language (Austin, 1962 and Searle, 1968), communicative action (Habermas, 1986), data modeling (Nijsen and Halpin, 1989) and ontologies (Guizzardi, 2005). In collaboration with universities in Antwerp (Belgium), Duisburg (Germany), Bolzano (Italy), Luxembourg, Vienna (Austria), Lisbon and Madeira (Portugal), Prague (Czech Republic), St. Gallen (Switzerland), San Jose (USA) and Tokyo (Japan), DEMO-4 was released in 2020. DEMO is managed by the Enterprise Engineering Institute.

and (re)designing organizations, in which human action is central. DEMO describes the structure of an organization, abstracted from the implementation. It describes what happens or should happen rather than how it happens. The focus is on the essence of an organization: WHAT the organization does, not HOW it does it. A DEMO model describes a partnership, either within an organization and across organizations. A DEMO model forms a consistent set of products, responsibilities, processes, information and business rules and gives an overview of, and insight into, organizations. It

4.4 Hierarchical structure of an interaction framework

An interaction framework incorporates all necessary digital interactions and roles to realize the purpose of a use case. [Figure 2](#) illustrates the hierarchy of definitions within this document. Each transaction defines a sequence of messages exchanged between two roles. A message is a digital form, including possible attachments, comprising data elements, which are categorized as complex and simple elements. The behaviour of simple elements is determined by its' user defined type and element conditions.

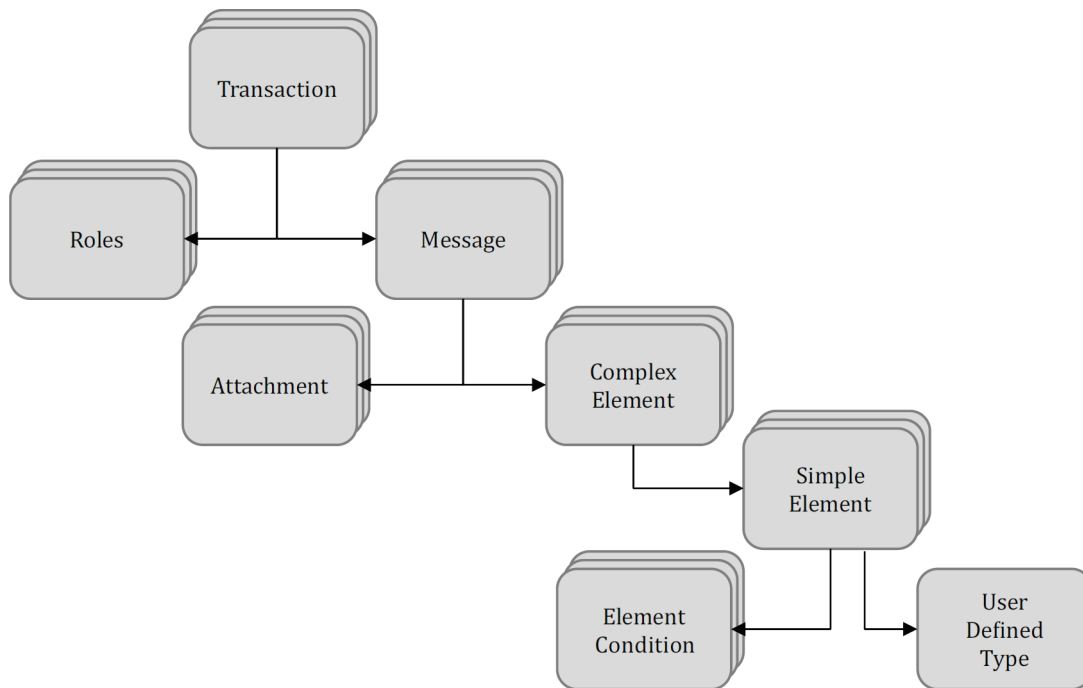


Figure 2 — Interaction framework with the hierarchy of its components

4.4.1 Roles

When defining a use case, roles are assigned to responsibilities and tasks, not specific functions or positions within an organisation. This is because the same roles consistently appear in business contexts, but individuals in different functions or positions may fulfil multiple roles or switch between them. For example, actors may fulfil roles such as 'initiator', 'client', 'builder', or 'designer'. The name of the role is derived from its primary activity, emphasizing the role's contribution within transactions.

A key principle is that people with roles act and negotiate with authority and responsibility. Only people can make agreements, take decisions, pass judgments, and enter into contracts. Roles therefor have to fulfil certain qualifications.

There is not always a one-to-one relationship between actors and roles. In some business processes, one actor may assume multiple roles, while in others, several actors may share the same role. For instance, a municipality could be both the initiator and the client, or a contractor for earthworks and an installation company might both take on the role of builder. An actor may also fulfil different roles across different business processes.

Every interaction framework includes defined roles to streamline responsibilities and tasks. This approach ensures clear and effective collaboration, providing a structured method to manage various functions efficiently within the business activity. Roles may also be reassigned if project members leave, and authorizations may be granted so others may take over their work. The audit trail always remains intact, allowing visibility about who did what and when.

4.4.2 Transactions

Each interaction framework consists of one or more transactions, this is defined by the use case and the associated business context. In an interaction framework a transaction is a sequence of messages between two roles. As shown in [Figure 3](#), each transaction has an initiator (the role that starts the transaction) and an executor (the role that carries out the transaction).

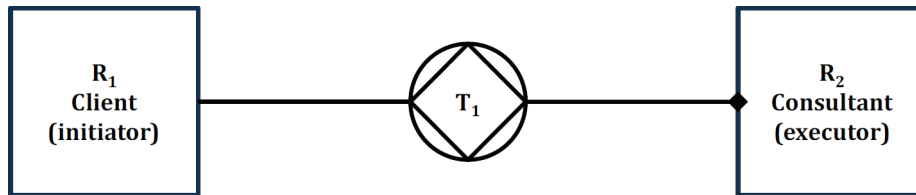


Figure 3 — A transaction always involves only two roles

Using transactions within information management brings information transfer into a process context. This ensures that information is exchanged systematically and purposefully within a structured digital process. Organizations using an interaction framework transition from document-driven to data-driven workflows by digitally managed information and processes.

4.4.2.1 Business process scenarios and sub transactions

Depending on the business context and use case, an interaction framework may contain one or several transactions. Within the business context sometimes multiple transactions between roles are required in sequence to achieve a goal. For example, when a role in transaction A prepares information that another role in transaction B shall review. In such cases, transactions need to be linked, forming a workflow known as a business process scenario.

In a business process scenario, there is always a main transaction where the initial request is made linked to one or more sub transactions. Sub transactions are designed to cascade from the main transaction. This is detailed in [Clause 5.2.2.10](#).

4.4.2.2 Interaction map

All necessary roles, transactions and sub transactions of a use case are described by the interaction map²⁾ in an Information Delivery Manual (IDM). This is an abstract representation that defines the boundary of a use case and highlights the essential communication chain required to fulfil its purpose. The name is derived from the fact that it records the interaction between the actors, i.e. their direct mutual influence. It leaves out all the activities and various process steps. The use of abstract roles makes the interaction map applicable to various situations. [Figure 4](#) shows a simplified example of an interaction map for a design office use case.

2) In DEMO terms the Interaction Map is called an Organization Construction Diagram (OCD), but for clarity purposes within the architecture, engineering and construction industry a different term is introduced by this document.

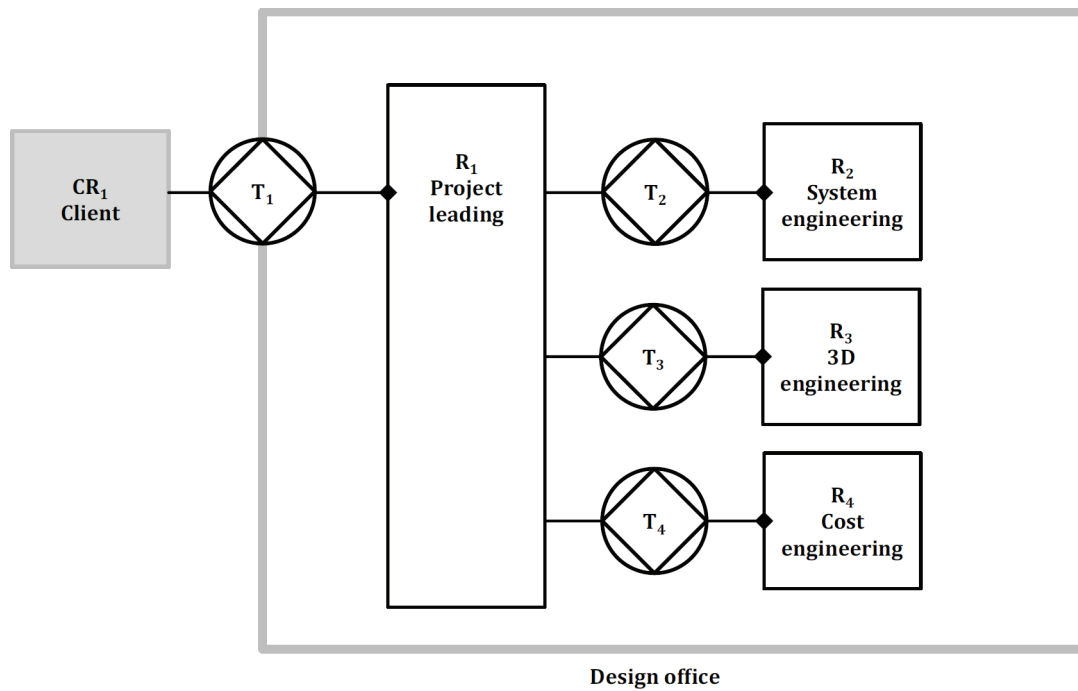


Figure 4 — Example of an interaction map of a design office use case

All roles and transactions in the interaction map have a unique identity and name, with numbering being arbitrary. Roles are named based on their primary activities, emphasizing their contributions to the use case. A composite role consists of multiple roles, but its composition is not relevant to the use case's scope.

4.4.2.3 Transaction product table

Each transaction has an intended result, also referred to as the product of that transaction. Together with the transactions and roles described by the interaction map, this can be summarized in a transaction product table. This table records the transaction kind (the name and unique number of each transaction), the product kind (the product of each transaction) and its initiating and executing roles. See [table 1](#) for an example.

Table 1 — Transaction product table of a simplified design office use case

Product kind	Transaction kind	Initiator role	Executor role
Design is delivered	T ₁ , Deliver design	CR ₁ Client	R ₁ Project leading
System specification is delivered	T ₂ , Deliver system specification	R ₁ Project leading	R ₂ System engineering
3D model is delivered	T ₃ , Deliver 3D model	R ₁ Project leading	R ₃ 3D engineering
Cost calculation is delivered	T ₄ , Deliver cost calculation	R ₁ Project leading	R ₄ Cost engineering

4.4.3 Messages

To realize the product of a transaction, interactions take place between two roles. In the digital world, these interactions are performed by means of digital messages. A transaction contains a set of messages that are exchanged for a specific purpose. The transaction also specifies the participating roles, the point in the lifecycle, and the sequence in which messages should be delivered (if applicable).

In terms of the IDM methodology, messages may carry information units and attachments that satisfy exchange requirements. Attachments may be of any type, including structured information containers such as IFC files or Information Containers (ICDD) and unstructured information containers such as documents, spreadsheets, pictures, video files and drawings. The interaction Framework message also supports the inclusion of metadata, such as version number, document date, and status. For certain message types, it may

be mandated that at least one attachment shall be added, and specific attachment types may be enforced by the interaction framework.

4.4.3.1 Sequence of messages

Sets of messages in a transaction form a chain with a beginning and an end. Most process diagrams provide a "happy flow", the best case scenario of communicative actions where both actors fulfil their role without dispute. However, in practice, processes rarely follow this ideal flow because humans can make mistakes so all possible outcomes should be considered, such as the need for adjustments or scenarios where a model is not approved.

For example, a transaction involving a request for a 3D model (see [Figure 5](#)) illustrates a message sequence in a transaction as a sequence diagram in UML notation. The transaction is initiated by R1 Project Leader with the message 'Request for 3D model.' The 3D engineer (role R3) responds with 'Work done and request for approval.' After a message 'Work approved' or 'Work not approved,' the transaction is completed.

The start message has no preceding message, being the only way to distinguish it from the other messages. Similarly, the final message has no following message. Each message has a direction indicated by the labelled arrow. A message may have several previous messages, e.g. a message approval may follow a delivery, but also a renewed delivery. Similarly, the final message has both a reject and an approval as the previous messages. Therefore, the chain of messages is not linear, but rather a mathematical graph.

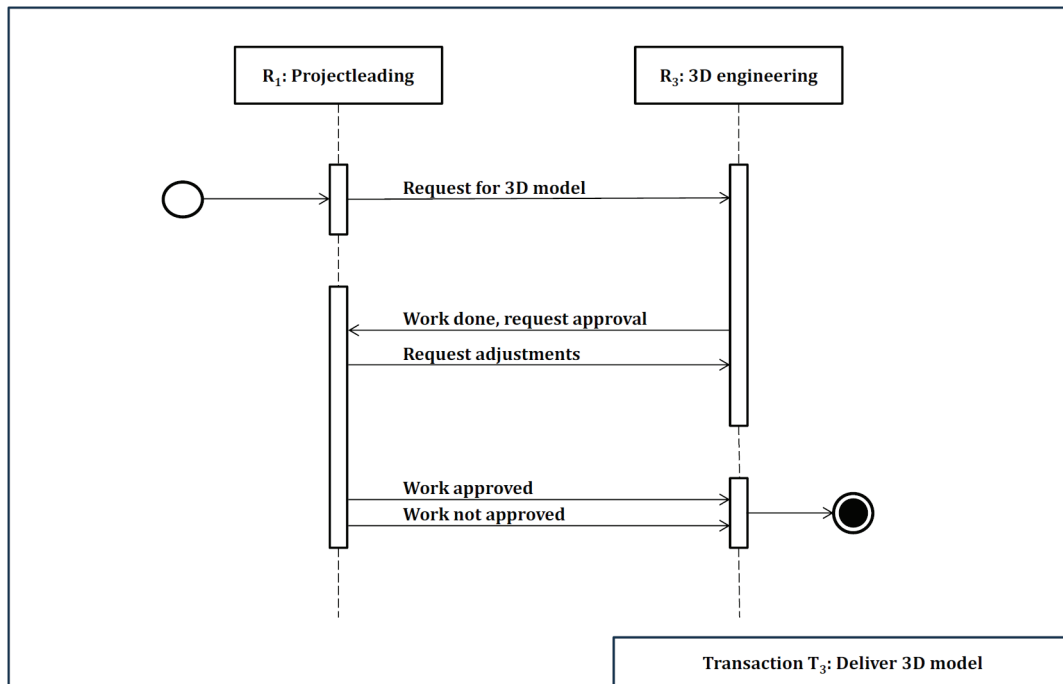


Figure 5 — Example of messages in a transaction

4.4.4 Complex and Simple Elements

A message is essentially a 'form' with data fields and attachments where required. Each message has a header with details of the involved roles and the date and time the message was sent. The body of the message includes the information to be transferred, such as: text fields, dates, amounts, Yes/No, list boxes, and tables. During digital IDM communication the data in the fields is stored in the database of the software that executes it.

Each data field in a message is called a "Simple Element", which is always part of a group called a "Complex Element". Each Complex Element may contain one or more Simple Elements. Simple Elements may be made mandatory or optional for users to fill in within a message. Each message shall contain at least one Complex

Element with at least one Simple Element, although a Simple Element may be empty. To create a table a Complex Element needs to be nested within another Complex Element.

The behaviour, restrictions (for example minimum length) and validation of Simple Elements are determined by their UserDefinedType. During digital IDM communication inheritance takes place. In principle, data entered in a Simple Element in a previous message will be sent in a subsequent message, unless a so-called Element Condition has been included in the interaction framework.

See [Figure 6](#) for an example of Complex and Simple Elements in the message “Proposal request for change.”

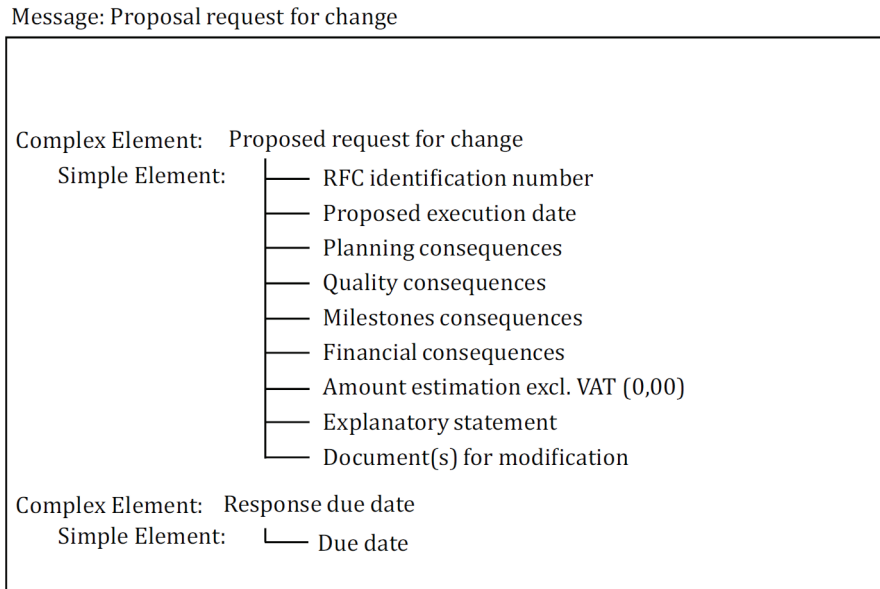


Figure 6 — Example of Complex and Simple Elements within a message

There is no rule about which data fields should be included in a message and which data should be in attachments created with other software tools. Depending on the level of detail needed for communicative decision-making, traceability and datamining purposes of the use case an interaction framework designer can decide to include or exclude data elements.

NOTE Camel case terms in this document denote XML definitions.

4.5 Establishing a Digital IDM communication Project

After the use case and the associated business context and exchange requirements are analysed, a digital IDM communication project can be set up. The scope of a digital IDM communication project equals the scope of the associated use case. The term project is explicitly not seen here as a construction term (e.g. the execution of a construction project), but as a generic term in which users can collaborate on a specific task. A software tool that works according to this document can support multiple projects, each containing an active interaction framework. Each project has a version history containing an active, as well as any possible previous interaction frameworks that were added, for example when a transaction was added, or the content of a message was changed. To avoid technical and legal problems, a transaction is always processed in the version of the interaction framework in which it was started.

4.5.1 Modelling the interaction framework

An interaction framework is essentially an XML (Extensible Markup Language) file, which allows data to be described in a specific computer interpretable structure. Since XML data is stored in plain text, XML is software and hardware independent and may therefore be used to exchange data between incompatible systems. To define how an interaction framework XML file should be set up an XSD (XML Schema Definition) file is used. This XSD file is named the interaction framework schema. The interaction framework schema determines the structure of an interaction framework XML file. It defines what elements and attributes may

or should be used, their relationship to each other and what types of data may be in them. The XSD may then be used by validation software to validate or proactively help create an interaction framework XML file.

The interaction framework schema serves two primary functions:

- To define the structure of the interaction framework XML file;
- To specify the elements, attributes, conditions and relationships within the interaction framework.

EXAMPLE The interaction framework schema defines that an interaction framework may include the definition of primary element types (e.g. SimpleElementType) and restrictions to primary element types (e.g. UserdefinedType) in an interaction framework.

In order to model an interaction framework efficiently in XML, the components described in the interaction framework schema such as TransactionTypes, MessageTypes, ComplexElementTypes, SimpleElementTypes may be used multiple times, to allow the same content to be used in multiple places. Interaction frameworks therefore often use references to the same components.

Each component contains a standard set of information, including ID and description. Transactions, messages, complex and simple elements may also contain help information that a compatible software tool can display to the user to help them better understand the purpose of the associated element.

There are specialised software tools that create and edit interaction frameworks. However, no specialised software is necessary to create an interaction framework. An interaction framework may be created with any tool that supports the creation of an XML file.

4.5.2 Creating and using the interaction framework schema

To determine if an interaction framework is correctly coded, it shall be validated. This validation is performed by a software tool that checks the interaction framework against the interaction framework schema.

To generate XML schema definitions, such as the interaction framework schema, this document uses EXPRESS source files (.EXP), as defined by ISO 10303-11. The EXPRESS file serves as a foundational model due to its formal and precise language, which captures complex relationships and constraints. It provides a clear, authoritative model that defines the data schema. From this model, various schema formats, including XSD, may be derived, ensuring a future proof method.

A key advantage of using EXPRESS is its ability to handle complex data models more efficiently and with fewer redundancies than XSD. In EXPRESS, data types and relationships are defined once and referenced throughout the schema. XSD does not have the same level of abstraction or reusability, resulting in repeated elements or data types, which makes an XSD schema challenging to maintain and prone to errors. When changes to the underlying data model or rules are necessary, updating the EXPRESS schema ensures an efficient approach and consistency across all other generated schemas, including the XSD.

To facilitate the conversion from EXPRESS to XSD, this document employs a generic algorithm, referred to as "Promotor" (see [Annex D](#)). This algorithm converts the EXPRESS schema into an XSD format that is compatible with broader web-based systems, while preserving the accuracy and complexity of the original model. This method ensures that the complex industry-specific rules are preserved during conversion, which would be difficult to achieve with XSD alone. This structured approach minimizes human error and ensures consistency across different data exchange scenarios.

[Clause 5](#) describes in detail how the interaction framework schema is created and specifies all available classes. Every interaction framework shall comply with this schema.

4.5.3 Using a project specific message to link people to roles

To initiate digital IDM communication, roles in the interaction framework shall be linked to specific actors (people) and their organizations. To create a secure, well-structured communication environment in supporting software, the interaction framework, organizations, and actors shall also be linked to a project with a unique ID and description. This is captured in a project specific message (PSM).

A digital IDM project therefore always consists of two elements: the interaction framework and the PSM. The interaction framework defines the overall communication structure, including roles, transactions, messages and elements. The PSM contextualizes this by linking the interaction framework to a specific project, indicating which use case the interaction framework represents and linking roles to participants from organizations. It specifies who may initiate or execute transactions and messages based on their role in the project.

Within the PSM, each person is associated with an organization and a role. Only those assigned specific roles may send messages. Multiple people may hold the same role, and one person may have several roles. The PSM also manages authorizations and successions, ensuring proper permissions for communication and handling role transitions smoothly during the project.

Additionally, the PSM implements the SOAP principle (see ISO 40210) to define the communication endpoint of an organisation. SOAP (Simple Object Access Protocol) provides a standardized format for message exchange between systems, using URLs to identify the endpoints for sending and receiving messages. In the PSM, each organisation is linked to a unique SOAP URL, ensuring that all communications are routed to the correct organisation in the project. This helps automate and secure the exchange of information between systems in line with the project's specific use case.

An example of a project specific message XML file can be found in [Annex E](#).

4.5.4 Executing digital IDM communication

Once a valid interaction framework and a project specific message are available, they may be interpreted by an information management software solution that is compatible to support digital IDM communication, such as a Common Data Environment (CDE). During digital IDM communication messages are exchanged between users of one or more software tools. Each software tool has its own SOAP endpoint where messages are sent to, in essence its digital mailbox. Depending on the security requirements and desires for interface with other applications, it may also be configured to give each organization within that software tool its own SOAP endpoint. Each message a user sends or receives is an XML file that includes a subject, a reference to the specific interaction framework version in which it was created and needs to be processed, the transaction it belongs to, the date and time it was sent, the route (from initiator to executor or vice versa), the names of the individuals with those roles, their respective organizations (along with their SOAP endpoint), any attachments, and the Complex and Simple Elements with their corresponding content. The structure, elements, attributes, conditions and relationships of a message XML are defined by the interaction framework and its interaction message schema XSD file. An example of a message XML file can be found in [Annex F](#).

A software solution that executes digital IDM communication facilitates at least the following requirements::

- Ensure the completeness and validity of an interaction framework;
- Facilitate the portability of an interaction framework;
- Facilitate updates of an interaction framework during execution;
- Facilitate the creation and updates of a project specific message in a project;
- Ensure the completeness and validity of messages that are sent and received;
- Ensure compatibility of communication between different software solutions with a communication protocol and encryption;
- Ensure verifiability by signing messages with advanced electronic signatures;
- Ensure digital IDM communication can be archived after a project is completed.

4.5.5 Validating digital IDM communication

When users from organizations in a use case start sending each other messages based on the interaction framework, it is essential to validate the received and sent messages to find corrupted messages, especially when they use different software tools.

To facilitate validation, a second XML Schema Definition shall be created. This schema definition is referred to as the interaction message schema. To generate this file, the rules from another EXPRESS source file, the “_5.EXP”, which contains the templates definition are used. Together with the validated interaction framework that is used to send that message a generic algorithm such as Promotor (see [Annex C](#)) uses the EXP_5 file (see [Annex B.2](#)) to create the interaction message schema specifically for that interaction framework.

EXAMPLE The ‘Promotor’ takes information from the interaction framework, such as the attribute CostEstimation to be used as a mandatory element in a specific message and creates an interaction defining the message with the CostEstimation attribute.

The validation process is asynchronous. This means that after receiving the attachments and the message in the receiving software, the validation does not have to take place immediately. The result of the validation is returned to the sending software with a reference to which the validation result belongs. The validation result has two options: no error (0), error (1). Optionally, additional text may be sent with the error code of the validation (0 or 1) to clarify the error. For example: that an attachment is missing, or the message does not conform to the project template (or framework), or the project is unknown.

Every information system that participates in the communication defined by the interaction framework should operate based on the corresponding interaction framework and interaction message schema. Every message sent or received should be valid according to the interaction message schema

4.5.6 Signing messages with advanced electronic signatures

A message that is created to be sent will be signed to ensure that its contents will remain verifiable in the future. A signed message shows that the content has not been altered and is the original content from the moment of sending. This ensures that the recipient did not tamper with a message. The signing will be sent with a message in such a way that signing is backwards compatible with non-signed messages. This is done by not including the signing in the decomposed message, but by attaching it after the decomposed message, while still in the body.

A software supplier is responsible for the management of the set of issued public keys it uses. At the same time each software supplier shall be able to produce a list of the issued certificates which includes the following:

- public key;
- public key friendly name; if available;
- corresponding server where the certificate is supposed to be used;
- corresponding validity start and end dates;
- Also, the software supplier shall be able to demonstrate that the private key of a public key is in their possession.

Key principles of message signing within digital IDM communication:

- Starting point for message exchange between the servers entails the following: all servers are known to each other and no server access without IP whitelisting is possible.
- This means that every software supplier knows which server pairs are communicating with each other. In case one the clients decides to host the software in their private cloud and as a result requires a unique public key, it is then the responsibility of the software supplier to provide their client with such public key.

- In the event of addition of the authentication layer in the future, the signing of the messages is not going to be affected. Only the ISO security level will change as a result. This in turn makes the central administration of the issued public keys unnecessary.
- Given the fact that communicating server pairs are known in advance, if it becomes apparent that a server of a software supplier A is using the public key of the software supplier B, then it is most certainly true that said suppliers have exchanged their private keys.

4.5.7 Changing the project during execution

Two types of changes may be made during the execution of a project. The first change concerns changes to the project, such as changed roles of people. For example, an extra project employee or a permanent replacement for a current employee. The second change concerns changes to the interaction framework, such as changed transaction type, changed message type.

Any change shall be implemented by all organizations involved. This means that the modified files (project specific message and/or interaction framework) shall be provided to all organizations involved in the project. Because changes can affect the content of a project, the content of the change shall be coordinated in advance between project members. To keep a valid audit trail a person may never be removed from the archived communication or project specific message as soon a message has been sent by that person.

4.5.8 Archiving digital IDM communication

After a project has been completed, the agreements and the messages are archived together. A software tool developed based on this document is expected to be able to read in a project so that a user can read the message traffic again and offer it as an offline project archive where the messages and appendices are preserved in the context of the workflow.

The exchange and archiving format for (completed) projects is referred to in this documentation as a "digital IDM communication archive". A digital IDM communication archive has the following requirements.

The information that shall be present in the archive includes all data from a maximum and a minimum of one project, namely:

- All framework versions used in the project;
- All project-specific messages from the project;
- All sent and received messages in the format xml, including the sequence in which they originated;
- All sent and received attachments in the original format.

Information from the archive shall be transparent and viewable without digital IDM communication software:

- Attachments are archived in the original format in which they were sent by the end user/organization. So a PDF file can be found as a PDF file. So archiving of attachments is not done in BASE64, MTOM or other similar formats.
- The archive shall have a structure so that the communication can be navigated in a logical manner.

The archive shall be importable:

- The information present in the archive shall be complete enough to reproduce/read the entire communication into a solution that implements this document.
- The connection between a message and an attachment shall be retrievable.
- After the import, a software tool shall be able to resume digital IDM communication within that project.

5 Schemas for validating interaction frameworks and messages

5.1 Introduction

In order to create valid interaction frameworks and validate messages during IDM digital communication, an interaction framework shall comply with the rules of the interaction framework schema and messages shall adhere to the rules of the interaction message schema.

Interaction frameworks and messages shall be validated with XSD files. Each XSD file shall be derived from an EXPRESS (EXP) source file. To generate the interaction framework schema, the EXPRESS source file “_2.EXP”, which contains the basic data model of the interaction framework, needs to be converted by a promotor to the “_3.XSD” file (see Annex A.3). During this conversion the rules in the “_2.EXP” source file are “promoted” to XSD classes. When this document refers to the interaction framework schema, it technically refers to the “_3.XSD” file. Because the basic components of the interaction framework have been established in this document, only one interaction framework schema XSD file is possible. If components are modified, added or removed in future versions of this document, a new version of interaction framework schema XSD file will be created. To verify that messages during digital IDM communication are structured according to the rules of the specific interaction framework within that project, a promotor algorithm generates an interaction message schema XSD based on the _5.EXP source file and the validated framework XML.

Figure 7 provides an overview of how a promotor algorithm uses the EXPRESS, XML and XSD files to validate an interaction framework and messages.

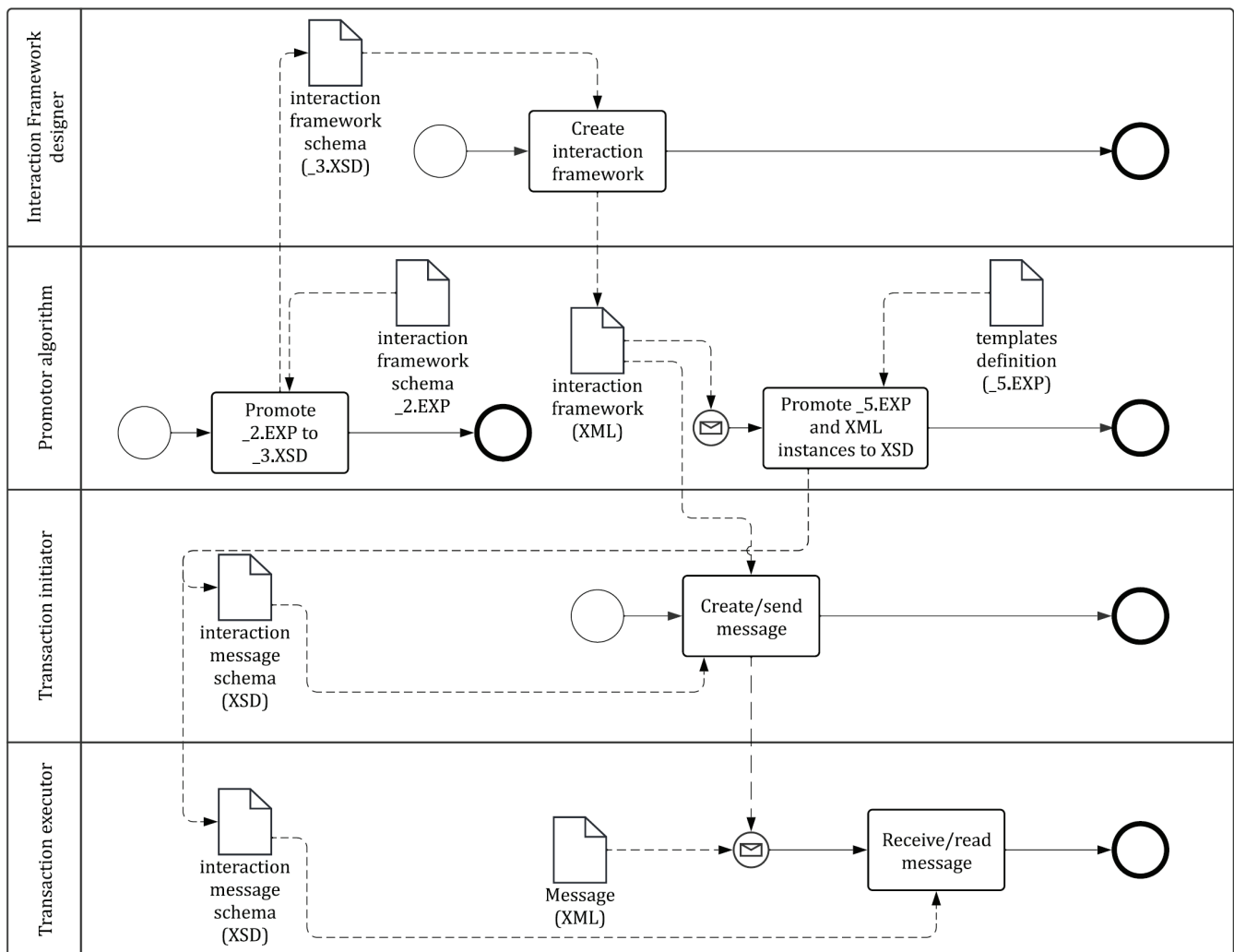


Figure 7 — How interaction frameworks and messages are validated

This clause describes the format of the interaction framework schema and the interaction message schema, as well as the application of all elements.

In relationship to [clause 5](#) the annexes will provide the complete schemas and examples. [Annex A](#) provides the full interaction framework schema, while [Annex B](#) provides the full interaction message schema. [Annex C](#) provides an example of a Promotor algorithm. [Annex D](#) may be used as an example of an instance of an interaction framework.

5.1.1 Basic principles

An interaction framework comprises instances of classes defined in the schema and shall be recorded as an XML file. Since an interaction framework is defined in XML, the word ‘type’ is used rather than class. [5.2](#) provides an overview of the element types that can occur in an interaction framework and are defined in the interaction framework schema.

Each schema is composed of two element types:

- Primary Element Types
- Secondary Element Types

The Primary Element Types are the basic building blocks of each interaction framework. Each Primary Element Type may contain one or more Secondary Element Types.

Each Primary Element Type shall include the attribute ‘ID’. This is a unique code within the interaction framework that provides a reference to this object instance within a given message. This reference may then be used multiple times.

Attribute ID is a specific definition in XML and is subject to more restrictions than Secondary Element Types. For example, an ID value may not contain spaces and often no special characters. Also, an ID value may not start with a number³⁾.

[Clause 5.2](#) describes the Primary and Secondary Element Types within the interaction framework schema, and [clause 5.3](#) describes the Primary and Secondary Element Types of the interaction message schema. Every description of a Primary Element Type includes a figure showing the corresponding Primary Element Type, the Secondary Element Types it contains and its references to other Primary Element Types. In addition, each unique reference between two Primary Element Types is explained and a figure has been added. For both schemas an inheritance reference matrix has been added showing the Primary Element Types each is Secondary Element Type is used in.

5.1.2 Element types in the interaction framework schema and _2 EXPRESS source file

5.1.3 Introduction

This clause lists and describes all the primary element types and secondary element types available in the _2 EXPRESS source file, which shall be used in combination with a promotor algorithm to create the interaction framework schema. An interaction framework shall be created from instances of these types and shall have a header which points to the schema with the defined available types.

[Figure 8](#) shows the model of the interaction framework schema, including all its types and references.

3) See [xml:id Version 1.0](<https://www.w3.org/TR/xml-id/>)

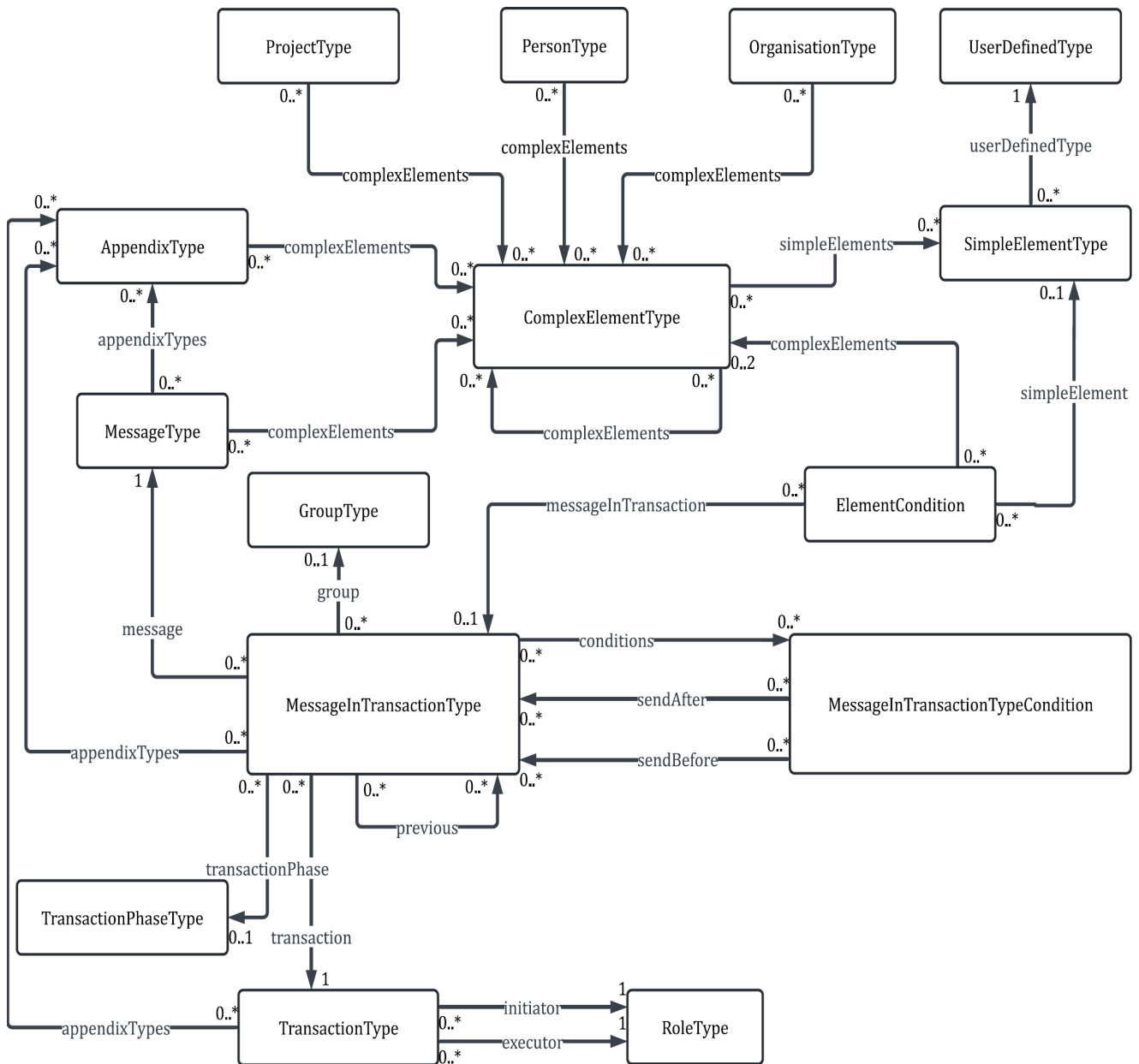


Figure 8 — Types and references of the interaction framework schema

5.1.4 Primary element types

5.1.4.1 ProjectType

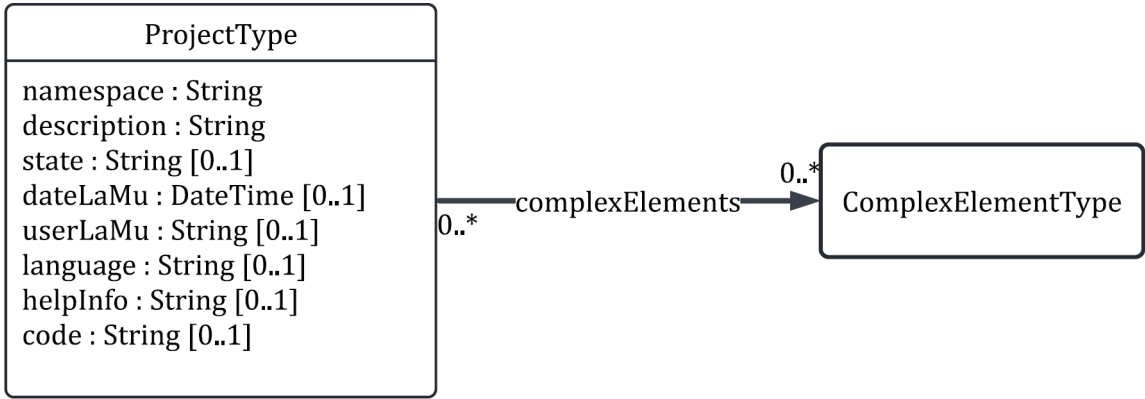


Figure 9 — ProjectType

An interaction framework shall always include one **ProjectType**. This is used to record the project name, interaction framework type and the framework namespace. Only one **ProjectType** is allowed for each interaction framework. The **ProjectType**, its secondary element types and possible references are displayed in [figure 9](#).

The basic fields of a **ProjectType**, such as the project name and project Id shall be stored in the project specific message and not included in the interaction framework.

5.1.4.2 PersonType

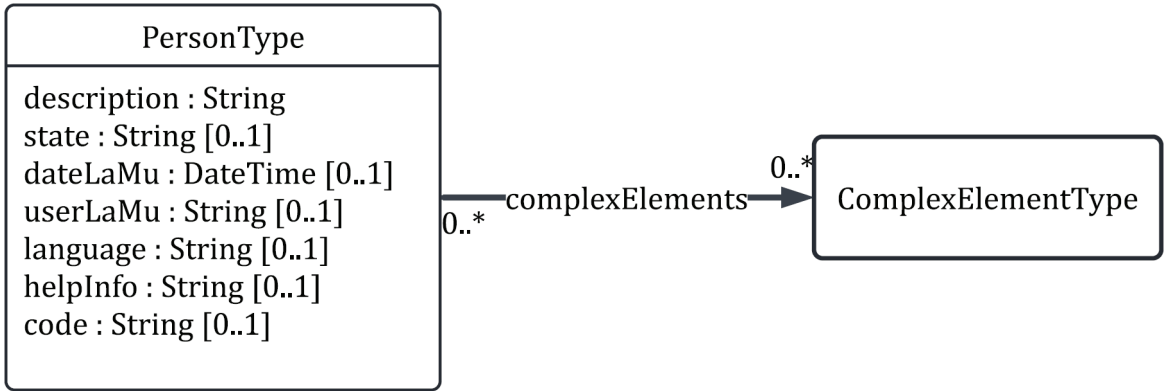


Figure 10 — PersonType

Every interaction framework shall contain one **PersonType**. This is used when creating people in the project specific message. The **PersonType**, its secondary element types and its reference to `complexType` are shown in [figure 10](#).

The basic fields of a person, such as name, username and Id abbreviation shall be stored in the project specific message and not in the interaction framework.

5.1.4.3 OrganisationType

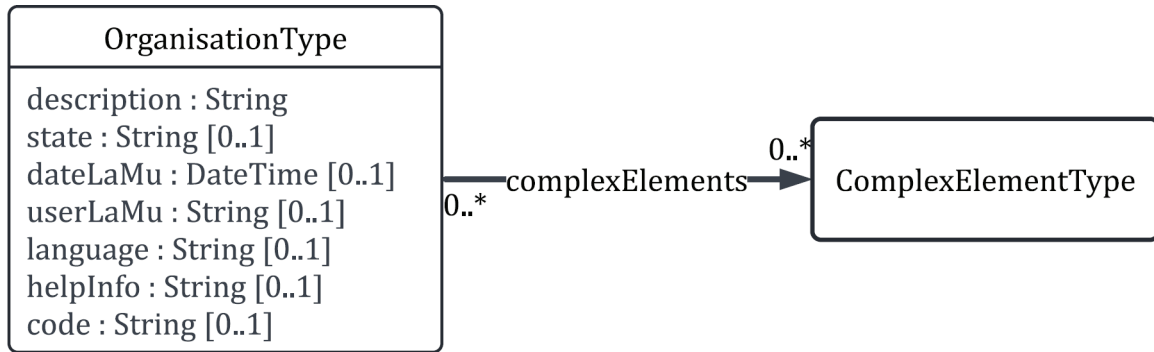


Figure 11 — OrganisationType

Every interaction framework shall include an OrganisationType. This is used when creating organizations in the project specific message. The OrganisationType, its secondary element types and its reference to complexElementType are displayed in [figure 11](#).

The basic fields of an organisation, such as name and Id abbreviation shall be stored in the project specific message and not in the interaction framework.

5.1.4.4 AppendixType

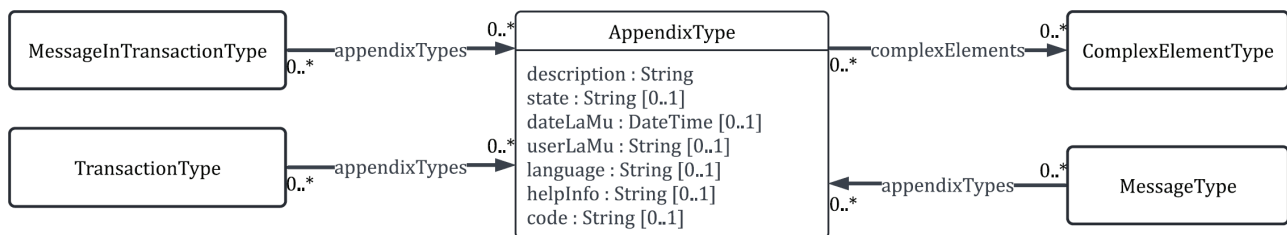


Figure 12 — AppendixType

The AppendixType is a definition to define the structure of elements regarding metadata of attachments. An instance of an AppendixType is used to define certain types of structured or unstructured information containers which can be part of the sent/received messages as attachments. The structure of the elements related to an instance of an AppendixType represents the specific metadata which is required for a certain type of attachment. [Figure 12](#) shows the AppendixType, its secondary element types and possible references.

Each message may have attachments. An uploaded file shall always be connected to an AppendixType. An interaction framework shall therefore always have at least one AppendixType. If an interaction framework has multiple AppendixTypes and no specific AppendixType is set on a message, the sender of that message may choose between the available AppendixTypes in the interaction framework to connect the attachment to that message.

An AppendixType is usually linked to a ComplexElement so that when entering an attachment, metadata fields are available to fill in associated data with the attachment.

5.1.4.5 ComplexElementType

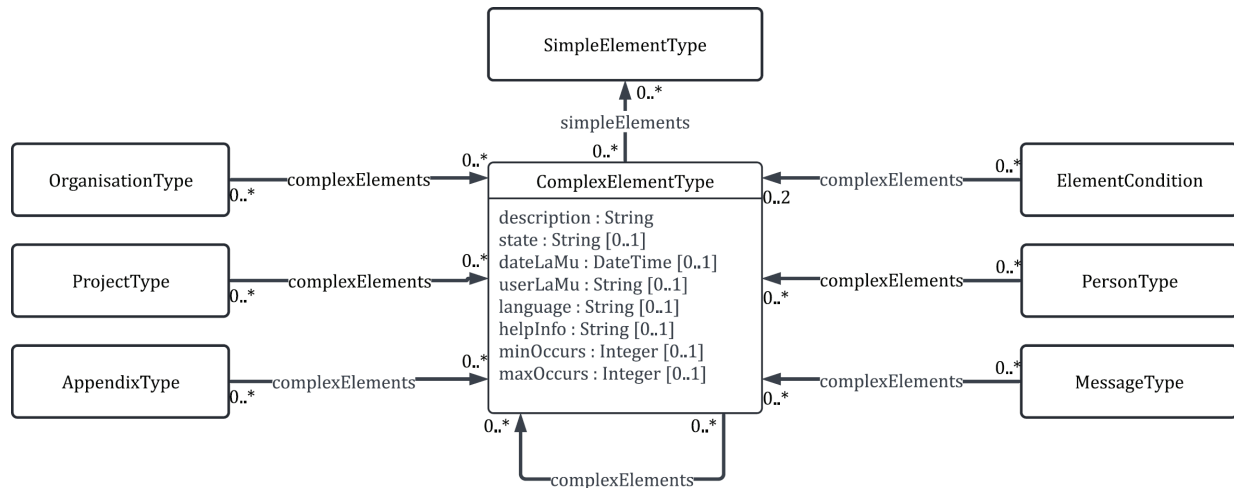


Figure 13 — ComplexElementType

Complex Elements (CE) can be seen as message chapters that are used to easily manage input fields (Simple Elements) in messages in an interaction framework and to make logical subdivisions in messages. Because only the different chapters are named in messages, a certain Simple Element may easily be added to or removed from an entire group of messages that contain that Complex Element. An overview of the ComplexElementType, its secondary element types and all possible references are presented in [figure 13](#).

The order in which the Simple Elements are stored in the Complex Element determines the order in which the compatible software displays these fields in the Complex Elements and in the messages.

The Complex Element description acts as a chapter title within messages so that it can be clearly explained what purpose that part of the message has. It can also explain the user which route the information takes or if information is kept within a certain domain.

5.1.4.5.1 Table Forming

In addition to adding Simple Elements to a Complex Element it is also permitted to name another Complex Element (child Complex Element) in a Complex Element (parent Complex Element). The agreed effect of this is that the child Complex Element is displayed in the parent Complex Element as table. The Simple Elements in the inserted child Complex Element will then be presented as columns.

This way, a Complex Element may consist of one or more tables, possibly also supplemented with normal message fields.

NB1: Currently it is technically not possible to indicate the order in which the tables in a Complex Element should be shown. As a result, a table is currently sometimes displayed above and sometimes below the other Simple Elements in that parent Complex Element, it is not possible to create a chapter in which fields and tables alternate.

NB2: Technically it is possible to create a chapter within a chapter within a chapter etc., which could for example produce a 3D table. There are no agreements on such applications and it has not yet been applied. This is considered to be room for unknown future applications.

5.1.4.6 MessageType

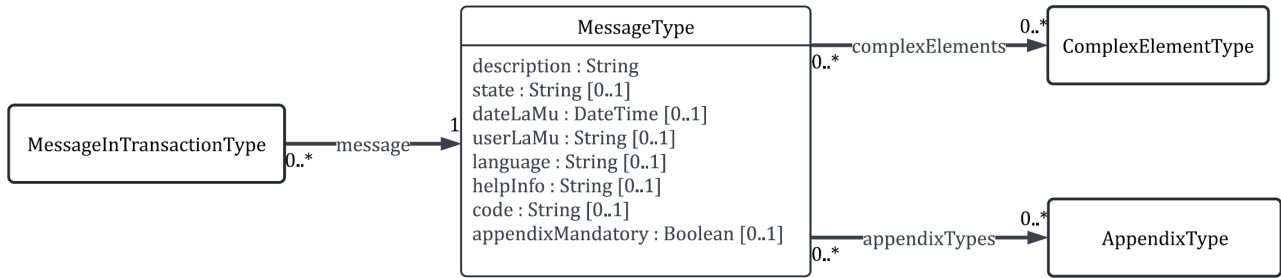


Figure 14 — MessageType

The MessageType defines which information may be exchanged. Within the execution of a transaction, the MessageType implicitly indicates the status of the transaction. The MessageType, its secondary element types and possible references are displayed in [figure 14](#).

5.1.4.7 Element Condition

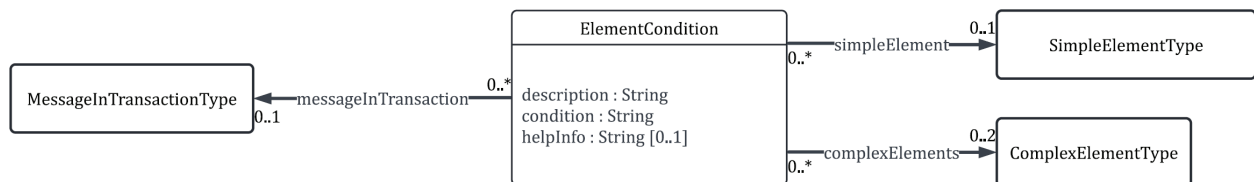


Figure 15 — Element Condition

An Element Condition controls if a user may or may not edit a field, or whether the field should be cleared by the software for a next message. The Element Condition, its secondary element types and possible references are displayed in [figure 15](#). The following conditions may be applied:

- FREE; this setting allows the editor of a reply or forward message of an existing message to edit this Simple Element. Compatible software copies the information from the handled existing message and is open for editing.
- EMPTY; this setting lets the compatible software present an empty and editable Simple Element, even if in the handled existing message already has information in this element.
- FIXED; this setting lets the compatible software present a copy of the information from the existing handled message where the editor does not have the possibility of editing the information. FIXED is the default setting for Simple Elements that already exist in the handled message, so it only has to be set to overrule a FREE or EMPTY condition with a lower priority (see [table 2](#)).

The basic rules of Element Conditions are:

- A Simple Element that is offered for the first time, which does not yet exist in the previous message, may always be filled in and is never blocked by this setting.
- By default, a Simple Element that already exists in the previous message is blocked in subsequent messages. Even if that field happened to be “empty” in the previous message. It may only be modified if the field is given the property “FREE”.
- The value of a Simple Element from a previous message is filled in the next message when it’s the same SimpleElementType and there is no condition ‘EMPTY’.
- Field restrictions may be applied to ComplexElementTypes, SimpleElementTypes and MessageInTransactionTypes, or a combination thereof.

— [Table 2](#) indicates which restriction to use if there are multiple restrictions on a field in a message:

Table 2 — Multiple restrictions decision matrix

	MessageIn-TransactionType	Regular or Parent ComplexElement-Type	Child ComplexElement-Type	SimpleElement-Type
0	-	-	-	-
1	-	-	-	X
2	-	-	X	-
3	-	-	X	X
4	-	X	-	-
5	-	X	-	X
6	-	X	X	-
7	-	X	X	X
8	X	-	-	-
9	X	-	-	X
10	X	-	X	-
11	X	-	X	X
12	X	X	-	-
13	X	X	-	X
14	X	X	X	-
15	X	X	X	X
16	If the begin message or element is not present in previous message then FREE.			

The highest number in the first column “wins”. So if a condition is set on a Complex Element (CE) and a Simple Element (SE) (see row 5) then this condition applies relative to a setting only on a SE (see row 1).

5.1.4.8 SimpleElementType

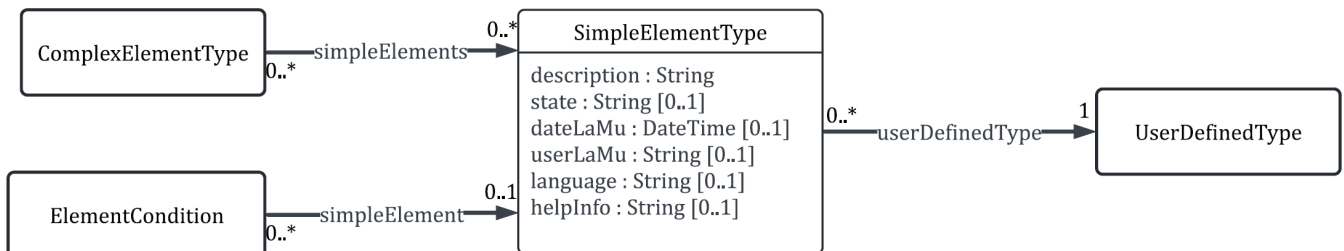


Figure 16 — SimpleElementType

Simple Elements are the actual data input fields within messages. The SimpleElementType, its secondary element types and its references to ComplexElementType, ElementCondition and UserDefinedType are shown in [figure 16](#).