

DIN EN ISO 29481-2



ICS 35.240.67; 91.010.01

**Bauwerksinformationsmodelle –
Handbuch der Informationslieferungen –
Teil 2: Interaktionsframework (ISO 29481-2:2012);
Deutsche Fassung EN ISO 29481-2:2016**

Building information models –
Information delivery manual –
Part 2: Interaction framework (ISO 29481-2:2012);
German version EN ISO 29481-2:2016

Modèles des informations de la construction –
Contrat d'interchange –
Partie 2: Cadre d'interaction (ISO 29481-2:2012);
Version allemande EN ISO 29481-2:2016

Gesamtumfang 94 Seiten

DIN-Normenausschuss Bauwesen (NABau)

Nationales Vorwort

Der Text von ISO 29481-2:2012 wurde vom Technischen Komitee ISO/TC 59/SC 13 „Organization of information about construction works“ der Internationalen Organisation für Normung (ISO) erarbeitet und als EN ISO 29481-2:2016 durch das Technische Komitee CEN/TC 442 „Building Information Modelling (BIM)“ übernommen, dessen Sekretariat vom SN (Norwegen) gehalten wird.

Als deutsches Spiegelgremium ist der Arbeitsausschuss NA 005-01-39 AA „BIM – Building Information Modelling (SpA zu ISO/TC 59/SC 13 – CEN/TC 442) des DIN-Normenausschusses Bauwesen (NABau) zuständig.

Für die im Dokument zitierte Internationale Norm wird im Folgenden auf die entsprechende deutsche Norm hingewiesen:

ISO 29481-1 siehe DIN EN ISO 29481-1

Aus Gründen der Praktikabilität sind die Anhänge A bis D von DIN EN ISO 29481-2 nicht ins Deutsche übersetzt worden.

Nationaler Anhang NA (informativ)

Literaturhinweise

DIN EN ISO 29481-1, *Bauwerksinformationsmodelle — Handbuch der Informationslieferungen — Teil 1: Methodik und Format (ISO 29481-1:2016)*

Deutsche Fassung

Bauwerksinformationsmodelle —
Handbuch der Informationslieferungen —
Teil 2: Interaktionsframework
(ISO 29481-2:2012)

Building information models —
Information delivery manual —
Part 2: Interaction framework
(ISO 29481-2:2012)

Modèles des informations de la construction —
Contrat d'interchange —
Partie 2: Cadre d'interaction
(ISO 29481-2:2012)

Diese Europäische Norm wurde vom CEN am 30. September 2016 angenommen.

Die CEN-Mitglieder sind gehalten, die CEN/CENELEC-Geschäftsordnung zu erfüllen, in der die Bedingungen festgelegt sind, unter denen dieser Europäischen Norm ohne jede Änderung der Status einer nationalen Norm zu geben ist. Auf dem letzten Stand befindliche Listen dieser nationalen Normen mit ihren bibliographischen Angaben sind beim CEN-CENELEC-Management-Zentrum oder bei jedem CEN-Mitglied auf Anfrage erhältlich.

Diese Europäische Norm besteht in drei offiziellen Fassungen (Deutsch, Englisch, Französisch). Eine Fassung in einer anderen Sprache, die von einem CEN-Mitglied in eigener Verantwortung durch Übersetzung in seine Landessprache gemacht und dem Management-Zentrum mitgeteilt worden ist, hat den gleichen Status wie die offiziellen Fassungen.

CEN-Mitglieder sind die nationalen Normungsinstitute von Belgien, Bulgarien, Dänemark, Deutschland, der ehemaligen jugoslawischen Republik Mazedonien, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, den Niederlanden, Norwegen, Österreich, Polen, Portugal, Rumänien, Schweden, der Schweiz, Serbien, der Slowakei, Slowenien, Spanien, der Tschechischen Republik, der Türkei, Ungarn, dem Vereinigten Königreich und Zypern.



EUROPÄISCHES KOMITEE FÜR NORMUNG
EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION

CEN-CENELEC Management-Zentrum: Avenue Marnix 17, B-1000 Brüssel

Inhalt

	Seite
Europäisches Vorwort	3
Vorwort	4
Einleitung	5
1 Anwendungsbereich	6
2 Normative Verweisungen	6
3 Begriffe	6
4 Grundlagen	7
4.1 Allgemeines	7
4.2 BIM und IDM	7
4.3 IDM-Komponenten	7
4.4 Grundlagen der betrieblichen Kommunikation	8
4.5 Interaktionsplan	9
4.6 Nachrichten in einer Transaktion	11
4.7 Interaktionsframework	12
4.8 Unterstützung von Softwarelösungen	12
5 Format eines Interaktionsframeworks	15
5.1 Einleitung	15
5.2 Informationstypen im Interaktionsframework-Schema	15
Anhang A (normativ) Interaction framework schema definition	18
Anhang B (normativ) Templates definition	56
Anhang C (informativ) Example interaction map of a simplified design office	76
Anhang D (informativ) Principles for Promotor algorithm	91
Literaturhinweise	92

Europäisches Vorwort

Der Text von ISO 29481-2:2012 wurde vom Technischen Komitee ISO/TC 59 „Buildings and civil engineering works“ der Internationalen Organisation für Normung (ISO) erarbeitet und als EN ISO 29481-2:2016 durch das Technische Komitee CEN/TC 442 „Building Information Modelling (BIM)“ übernommen, dessen Sekretariat vom SN gehalten wird.

Diese Europäische Norm muss den Status einer nationalen Norm erhalten, entweder durch Veröffentlichung eines identischen Textes oder durch Anerkennung bis April 2017, und etwaige entgegenstehende nationale Normen müssen bis April 2017 zurückgezogen werden.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. CEN [und/oder CENELEC] sind nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Entsprechend der CEN-CENELEC-Geschäftsordnung sind die nationalen Normungsinstitute der folgenden Länder gehalten, diese Europäische Norm zu übernehmen: Belgien, Bulgarien, Dänemark, Deutschland, die ehemalige jugoslawische Republik Mazedonien, Estland, Finnland, Frankreich, Griechenland, Irland, Island, Italien, Kroatien, Lettland, Litauen, Luxemburg, Malta, Niederlande, Norwegen, Österreich, Polen, Portugal, Rumänien, Schweden, Schweiz, Slowakei, Slowenien, Spanien, Tschechische Republik, Türkei, Ungarn, Vereinigtes Königreich und Zypern.

Anerkennungsnotiz

Der Text von ISO 29481-2:2012 wurde vom CEN als EN ISO 29481-2:2016 ohne irgendeine Abänderung genehmigt.

Vorwort

ISO (die Internationale Organisation für Normung) ist eine weltweite Föderation von nationalen Normungsorganisationen (ISO Mitglieder). Die Erstellung von Internationalen Normen wird normalerweise von ISO Technischen Komitees durchgeführt. Jede Mitgliedsorganisation, das Interesse an einem Thema hat, für welches ein Technisches Komitee gegründet wurde, hat das Recht, in diesem Komitee vertreten zu sein. Internationale Organisationen, staatlich und nicht-staatlich, in Liaison mit ISO, nehmen ebenfalls an der Arbeit teil. ISO arbeitet eng mit der Internationalen Elektrotechnischen Kommission (IEC) bei allen elektrotechnischen Themen zusammen.

Internationale Normen werden in Übereinstimmung mit den Regeln nach ISO/IEC Direktive, Teil 2 erarbeitet.

Die Hauptaufgabe von Technischen Komitees ist es, Internationale Normen zu erarbeiten. Internationale Norm-Entwürfe, die von Technischen Komitees verabschiedet wurden, werden den Mitgliedsorganisationen zur Abstimmung zur Verfügung gestellt. Für die Veröffentlichung als Internationale Norm werden mindestens 75 % Zustimmung der Mitgliedsorganisationen benötigt.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Teils von ISO 29481 Patentrechte berühren können. ISO ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

ISO 29481-2 wurde vom Technischen Komitee ISO/TC 59 „Buildings and civil engineering works“, Unterkomitee SC 13 „Organization of information about construction works“, erarbeitet.

ISO 29481 besteht aus folgenden Teilen mit dem Haupttitel *Building information models — Information delivery manual*:

- *Part 1: Methodology and format*
- *Part 2: Interaction framework*

Die folgenden Teile sind in Vorbereitung:

- *Part 3: Model view definitions*

Einleitung

Die Bauwerksinformationsmodellierung bietet ein Konzept für die Beschreibung und Darstellung von Informationen, die für die Planung, den Bau und den Betrieb von baulichen Anlagen erforderlich sind. Sie ermöglicht die Zusammenführung verschiedener Informationen, die im Bauwesen verwendet werden, in einer einheitlichen Informationsumgebung und reduziert oder beseitigt oftmals sogar die Notwendigkeit von Papierunterlagen.

Ein Handbuch der Informationslieferungen (IDM, en: information delivery manual) ist ein wichtiges Hilfsmittel, um den vollen Nutzen aus einem Bauwerksinformationsmodell (BIM, en: building construction information model) zu ziehen. Sind die erforderlichen Informationen vorhanden und qualitativ zufriedenstellend, kann der eigentliche Bauprozess erheblich verbessert werden. Um dies zu erreichen, sollte ein gemeinsames Verständnis aller Bauprozesse vorliegen, einschließlich der für die Ausführung benötigten Informationen sowie der Ergebnisse der Ausführung.

Der Fokus dieses Teils von ISO 29481 liegt auf den Aspekten des Bauprozesses, die sich auf die Lenkung und die Koordinierung der Beteiligten beziehen. Die Koordinierung hängt von der Kommunikation ab, die gut strukturiert, eindeutig, detailliert und zeitnah sein sollte. Auf Grund des Schwerpunkts Koordinierung und Interaktion stellt dieser Teil von ISO 29481 eine natürliche Ergänzung zu Normen wie ISO 10303-239 und ISO 16739 dar, die die Bauwerksmodellierung als Schwerpunkt haben.

Dieser Teil von ISO 29481 legt eine Methodik und ein Format für die Darstellung der Koordinierungsaktivitäten der an einem Bauprojekt beteiligten Akteure dar. Darin wird beschrieben, wie die Koordinierungsprozesse und die zur Durchführung dieser Prozesse erforderlichen Informationen zu identifizieren und zu definieren sind. Die sich daraus ergebenden Interaktionsstrukturen ermöglichen die Standardisierung von Interaktionen in den Bauprozessen auf nationaler und lokaler Ebene sowie auf Projektebene. Dieser Teil der Norm enthält ebenfalls ein Format für die Unterstützung von Lösungen, die durch IKT-Lösungsanbieter zur Verfügung gestellt werden. Die Unterstützung verschiedener IKT-Lösungen, die durch diesen Teil von ISO 29481 bereitgestellt wird, bedeutet, dass verschiedene Systeme für das Prozessmanagement zusammengeführt werden. Dadurch bietet dieser Teil von ISO 29481 Anwendern eine Grundlage für den verlässlichen Austausch und das Teilen von Informationen, damit sie sicher sein können, dass die Informationen, die sie senden oder empfangen, korrekt und für die von ihnen durchzuführenden Koordinierungsaktivitäten ausreichend sind.

Anlass der Entwicklung dieses Teils von ISO 29481, der vornehmlich auf der 2003 erarbeiteten niederländischen VISI-Norm beruht, war das Bedürfnis der Anwender nach Verlässlichkeit im Informationsaustausch.

1 Anwendungsbereich

Dieser Teil von ISO 29481 legt eine Methodik und ein Format zur Beschreibung der „Koordinierungsaktivitäten“ der an Bauprojekten Beteiligten während aller Lebenszyklusphasen fest.

Er legt daher Folgendes fest:

- eine Methodik zur Beschreibung eines Interaktionsframeworks;
- eine geeignete Methode, Verantwortlichkeiten und Interaktionen darzustellen, die einen Prozesskontext für den Informationsfluss zur Verfügung stellt;
- ein Format, in dem das Interaktionsframework festgelegt werden sollte.

Dieser Teil von ISO 29481 soll die Interoperabilität von Softwareprogrammen, die während des Bauprozesses eingesetzt werden, erleichtern, die digitale Zusammenarbeit verschiedener Akteure im Bauprozess fördern und eine Grundlage für einen fehlerfreien, verlässlichen, wiederholbaren und qualitativ hochwertigen Informationsaustausch schaffen.

2 Normative Verweisungen

Die folgenden Dokumente, die in diesem Dokument teilweise oder als Ganzes zitiert werden, sind für die Anwendung dieses Dokuments erforderlich. Bei datierten Verweisungen gilt nur die in Bezug genommene Ausgabe. Bei undatierten Verweisungen gilt die letzte Ausgabe des in Bezug genommenen Dokuments (einschließlich aller Änderungen).

ISO 29481-1, *Building information models — Information delivery manual — Part 1: Methodology and format*

3 Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

3.1
Handbuch der Informationslieferungen
IDM
(en: information delivery manual)
Dokumentation (Handbuch), das die betrieblichen Prozesse und die detaillierten Spezifikationen bezüglich der Informationen, die ein Beteiligter entsprechend seiner Rolle zu einem bestimmten Zeitpunkt in einem Projekt liefern muss, beschreibt

3.2
Interaktionsframework
formale Festlegung, einschließlich der Definition der Rollen, der Transaktionen, von Transaktions-Nachrichten und der Daten-Elemente in diesen Nachrichten

3.3
Interaktionsframework-Schema
formale Festlegung der Regeln, die im Interaktionsframework eingehalten werden müssen

3.4
Interaktionsschema
formale Beschreibung der Regeln, die gesendete und empfangene Nachrichten einhalten müssen

3.5

Promotor

Algorithmus, der ein Interaktionsschema generiert aus einem Interaktionsframework, einem Interaktionsframework-Schema und einer Vorlagensammlung als Eingabedaten

3.6

Vorlagensammlung

eine Anzahl von Vorlagen enthält, die unabhängig vom Interaktionsframework sind, zur Generierung eines Interaktionsschemas

3.7

VISI

Akronym für eine niederländische Norm für die Kommunikation zwischen den an Bauvorhaben Beteiligten

Anmerkung 1 zum Begriff: VISI steht für „Voorwaarden scheppen voor Invoeren Standaardisatie ICT in de Infrastructuursector“, d. h. „Erarbeitung von Bedingungen für die Umsetzung der Normung im IKT-Bereich für den Bausektor“.

4 Grundlagen

4.1 Allgemeines

Dieser Abschnitt wurde hinzugefügt, um wichtige Konzepte hervorzuheben und zu erläutern, auf denen dieser Teil von ISO 29481 basiert.

4.2 BIM und IDM

Bauwerksinformationsmodelle fassen die unterschiedlichen Informationen, die im Rahmen eines Bauprojekts verwendet werden, in einer gemeinsamen Informationsumgebung zusammen. Damit dies möglich ist, ist ein gemeinsames Verständnis der Bauprozesse und der Informationen erforderlich, die für die Durchführung benötigt werden.

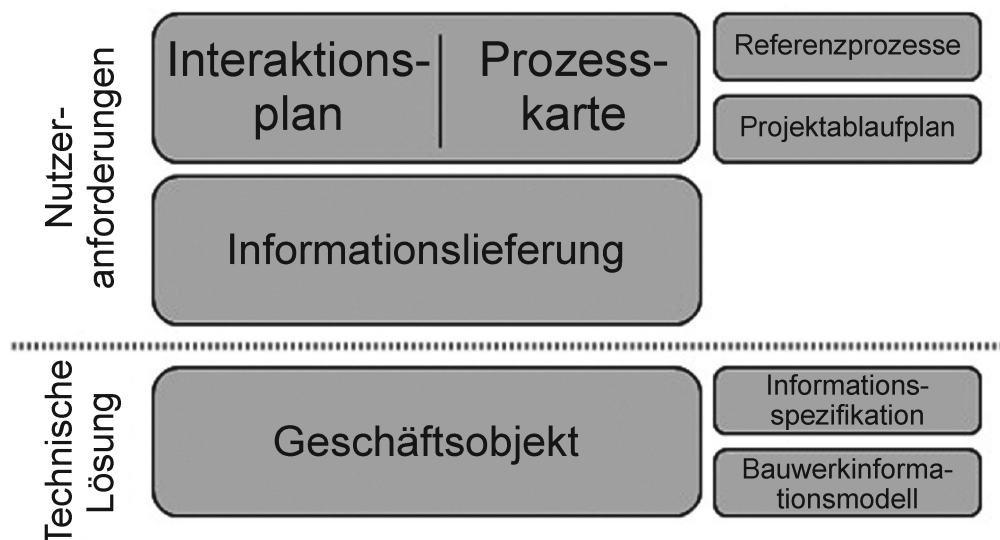
Bei der ISO 29481 handelt es sich um eine Norm, die eine Methode zur Entwicklung eines Handbuchs der Informationslieferungen beschreibt.

Die in ISO 29481-1 beschriebene IDM-Methoden sind die Basis zur Entwicklung und Verwendung von IDM.

4.3 IDM-Komponenten

Die IDM-Methoden und -Komponenten werden in ISO 29481-1 beschrieben. In diesem Abschnitt wird eine Illustration zur Verfügung gestellt, die grafisch darstellt, welche unterschiedlichen IDM-Komponenten vorhanden sind und wie diese in Zusammenhang stehen.

Innerhalb eines IDM gibt es zwei Perspektiven. Dabei handelt es sich um Nutzeranforderungen und technische Lösungen. Innerhalb der zwei Perspektiven gibt es mehrere Zonen, durch welche die unterschiedlichen IDM-Komponenten charakterisiert werden (siehe Bild 1).

**Bild 1 — IDM-Zonen**

Innerhalb der Nutzeranforderungsperspektive sind diese Zonen

- Interaktionspläne, mit denen die Rollen und die Interaktionen der Rollen beschrieben werden,
- Prozesskarten, mit denen der Gesamtprozess beschrieben wird, über den Informationen ausgetauscht werden,
- Informationslieferung, womit der erforderliche Informationsaustausch beschrieben wird,
- Referenzprozesse (gespeicherte Austauschbeschreibungen),
- der Projektablaufplan (Auftreten von Prozessen im Rahmen eines Projekts).

Die Perspektive für technische Lösungen umfasst

- die Geschäftsobjekte, die das Austauschansforderungsmodell umfassen,
- die Informationsspezifikation, womit das Schema beschrieben wird, auf dem der Informationsaustausch basiert,
- das Bauwerksinformationsmodell.

Dieser Teil von ISO 29481 beschäftigt sich insbesondere mit den Interaktionsplänen und basiert auf den allgemeinen Prinzipien der betrieblichen Kommunikation.

4.4 Grundlagen der betrieblichen Kommunikation

Sobald ein Kunde die Lieferung eines Produkts oder einer Dienstleistung angefragt hat, wird eine Kette von Aktivitäten ausgelöst, die zur Bereitstellung des Produkts oder der Dienstleistung führt. Eine solche Kette von Aktivitäten wird Geschäftsprozess genannt. In diesem Fall wird auch von einem primären Geschäftsprozess gesprochen, da er extern initiiert wird.

Die Kommunikation zwischen den beteiligten Parteien stellt einen Teil des Geschäftsprozesses dar. Dieser Teil von ISO 29481 beschäftigt sich insbesondere mit der Kommunikation in Bezug auf die Lieferung eines Ergebnisses (performative Kommunikation). Eine Anforderung wird durch Kommunikationsaktionen initiiert und ausgeführt. An einer Kommunikationsaktion sind immer zwei Parteien beteiligt: die Person, die die Aktion ausführt und die Person, an die sich die Aktion richtet. Die Handhabung einer Anforderung tritt in einem bestimmten Muster auf, das Transaktion genannt wird.

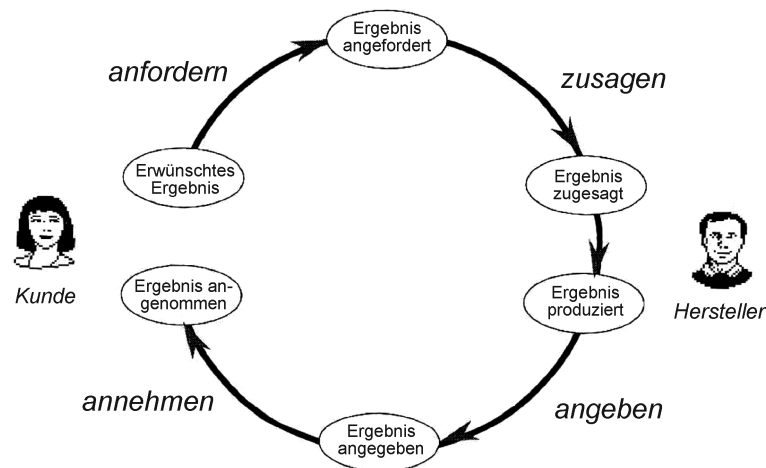


Bild 2 — Transaktionsmuster (Dietz, 2006)

In Bild 2 wird die einfachste Form dieses Transaktionsmusters dargestellt. Darin wird gezeigt, dass zur Erbringung eines Produktionsergebnisses (z. B. kann das „erwünschte Ergebnis“ die Lieferung eines Dokuments sein) dieses Ergebnis zuerst durch eine Person, die die Rolle des Kunden hat, von einer Person in der Rolle des Herstellers angefordert werden muss. Damit geht der Prozess in den Zustand „Ergebnis angefordert“ über. Der Hersteller antwortet auf die Anforderung mit der Zusage, das gewünschte Ergebnis zu produzieren, wodurch der Prozess in den Zustand „Ergebnis zugesagt“ übergeht. Der Hersteller hat nun also eine Aufgabe auszuführen: Er muss die Zusage einhalten, indem er das Dokument erstellt und es an den Kunden liefert. Nach der Übergabe des Dokuments an den Kunden gibt er an, dass er die Zusage eingehalten hat. Der Kunde reagiert auf diesen Zustand, indem er das Ergebnis als produziert annimmt. Mit dieser Handlung ist die Transaktion abgeschlossen.

Bei der Ausführung eines solchen Geschäftsprozesses sind häufig mehrere Personen beteiligt. Ihr Verhalten hängt von ihrer Rolle bei diesem Prozess ab. Rollen/Teilnehmer führen durch das Abschließen von Transaktionen Geschäfte mit anderen Rollen/Teilnehmern durch. Eine nützliche Darstellung dieser Interaktion zwischen den Rollen/Teilnehmern wird Interaktionsplan genannt.

4.5 Interaktionsplan

Ein Interaktionsplan legt die relevanten Rollentypen und Transaktionstypen für einen bestimmten Prozess fest. IDM unterscheidet zwischen einer Rolle, die eine Anforderung sendet, dem Initiator, und der Rolle, die auf diese Anforderung reagiert, dem Ausführer. Eine Transaktion darf nur eine initiiierende Rolle und eine ausführende Rolle besitzen. Bild 3 zeigt die Komponenten des Interaktionsplans.

ANMERKUNG Die im Interaktionsplan verwendeten Bezeichnungen basieren auf dem Konstruktionsmodell, das in den Publikationen von Prof. Jan L.G. Dietz beschrieben wird. Diese Bezeichnungen unterscheiden sich von den BPMN-Bezeichnungen und werden zur Erstellung von möglichst einfachen Karten verwendet. Hierbei wird zudem das Konzept einer „Transaktion“ verwendet, das in BPMN nicht vorhanden ist.

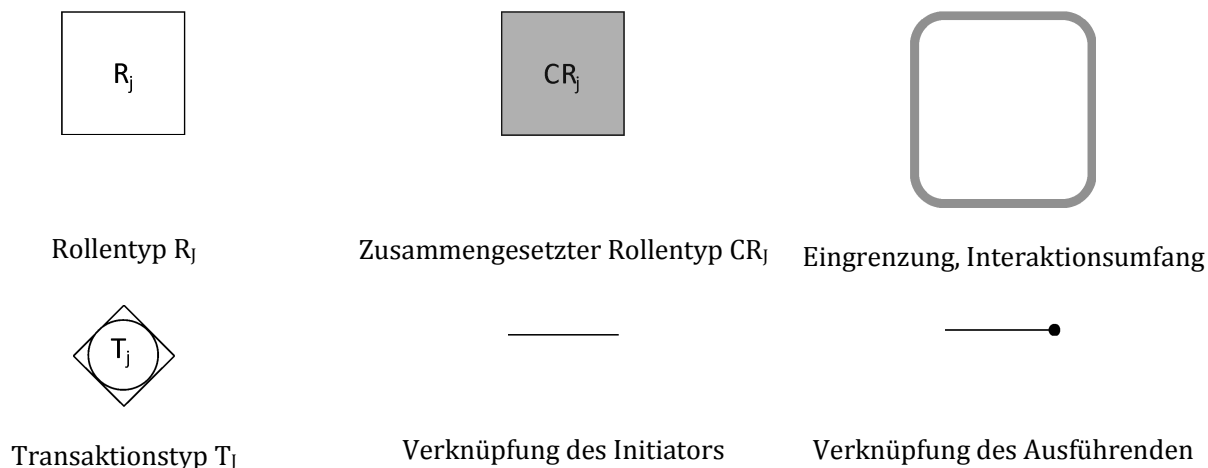


Bild 3 — Komponenten eines Interaktionsplans

Der Vorteil des Interaktionsplans ist, dass das Augenmerk auf die Schnittstellen zwischen den Rollen gelenkt wird und die Komplexität des Prozesses innerhalb der Rollendomäne sowie die Details der Interaktion zwischen den Rollen nicht gezeigt werden. Die Verwendung abstrakter Rollen führt dazu, dass der Interaktionsplan in vielen unterschiedlichen Situationen eingesetzt werden kann. Der Interaktionsplan ist ein nützliches Hilfsmittel zur Analyse und Festlegung wesentlicher Elemente eines Geschäftsprozesses. Bild 4 zeigt ein vereinfachtes Beispiel eines Interaktionsplans für ein Designbüro.

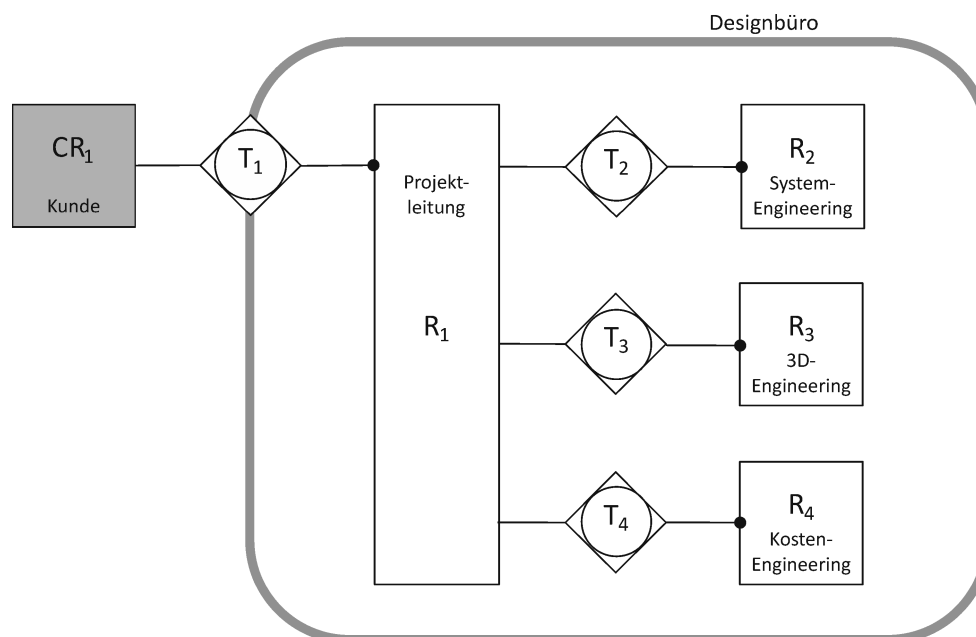


Bild 4 — Beispiel eines Interaktionsplans

In einem Interaktionsplan müssen alle Transaktionen enthalten sein, die für die Handhabung der erforderlichen Beiträge der relevanten Rollen zu BIM erforderlich sind. Alle Rollen und Transaktionen im Interaktionsplan müssen eine eindeutige Identität und eine eindeutige Bezeichnung besitzen. Die Nummerierung kann beliebig erfolgen. Die Bezeichnung der Rolle basiert auf der Haupttätigkeit, die von dieser Rolle ausgeführt wird. Damit wird der Beitrag dieser Rolle zu BIM verdeutlicht. Eine Mischrolle ist eine Rolle, die Elemente mehrerer Rollen in sich vereint, deren Zusammensetzung jedoch unbekannt oder nicht relevant ist.

Die Interaktionen können in einer Transaktionstabelle zusammengefasst werden.

Tabelle 1 — Transaktionstabelle eines vereinfachten Designbüros

Transaktionsergebnis	Transaktionstyp
Entwurf geliefert	T ₁ , Entwurf liefern
Systemspezifikation geliefert	T ₂ , Systemspezifikation liefern
3D-Modell geliefert	T ₃ , 3D-Modell liefern
Kostenkalkulation geliefert	T ₄ , Kostenkalkulation liefern

4.6 Nachrichten in einer Transaktion

Eine Transaktion muss mehrere Nachrichten enthalten, die zu einem bestimmten Zweck ausgetauscht werden. Die Transaktion legt zudem die teilnehmenden Rollen, Punkte im Lebenszyklus und die Reihenfolge fest, in der Nachrichten zu liefern sind (falls zutreffend).

Ein Beispiel für eine Transaktion ist die Bearbeitung einer Anforderung für ein 3D-Modell. In Bild 5 werden die Nachrichten in einer Transaktion als Abfolgediagramm mit UML-Bezeichnungen dargestellt. Die Transaktion kann nur von „R1 Projektleitung“ mit der Nachricht „Anforderung eines 3D-Modells“ initiiert werden. Der 3D-Ingenieur (Rolle R3) kann mit der Nachricht „Beendigung der Arbeit und Anforderungsbestätigung“ antworten. Nachdem die Nachricht „Abnahme der Arbeit“ oder „Ablehnung der Arbeit“ gesendet wurde, ist die Transaktion abgeschlossen.

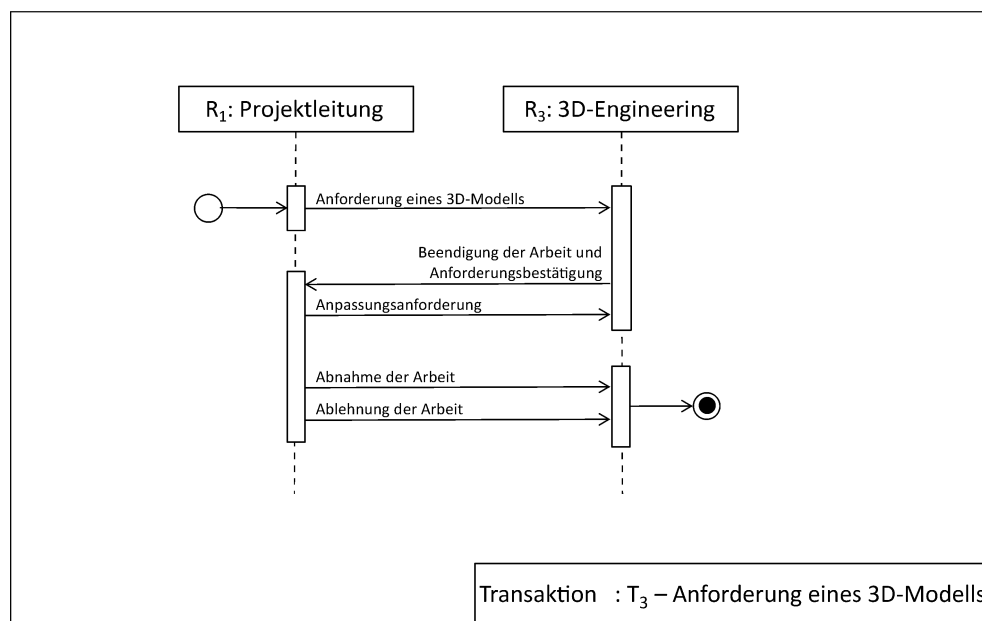


Bild 5 — Beispiele für Nachrichten in einer Transaktion

Eine Nachricht ist ein Informationsmodell und enthält Daten. Die Nachrichten können Anhänge enthalten. Als Anhang kann eine Austauschforderung an die ausführende Rolle übermittelt werden und das Ergebnis (Beitrag zum BIM) wird an die initiiierende Rolle übermittelt. Durch die Verwendung von Transaktionen wird der Informationstransfer in einen Prozesskontext gesetzt.

4.7 Interaktionsframework

Um als Richtlinie für einen Prozess und Informationstransfer zu dienen, müssen die Interaktionselemente auf verständliche Art und Weise beschrieben werden. Diese verständliche Beschreibung wird als Interaktionsframework bezeichnet. Ein Interaktionsframework hat Folgendes zu enthalten:

- Definition relevanter Rollen,
- Transaktionen,
- Nachrichten in einer Transaktion,
- die Abfolge der Nachrichten in einer Transaktion,
- Datenelemente in Nachrichten.

Ein Interaktionsframework kann für einen festgelegten Anwendungsbereich vorbereitet und als Standard auf einer (inter)nationalen Ebene, Organisationsebene oder Projektebene verwendet werden. Beispielsweise wird in den Niederlanden ein Interaktionsframework zur Durchführung aller vertraglichen Belange während der Ausführung eines Bauvorhabens auf nationaler Ebene entwickelt. Dieser Teil von ISO 29481 wird von Organisationen und Projekten als Vorlage verwendet und an die individuellen Bedürfnisse angepasst.

BEISPIEL Ein Interaktionsframework kann die Attribute „CostEstimation“ als Instanz von „SimpleElementType“ umfassen, welches als verpflichtendes Element einer bestimmten Nachricht verwendet wird. Sie kann zudem eine Einschränkung bezüglich des Formats des Attributs „CostEstimation“ enthalten (z. B. nur Euro-Werte mit zwei Dezimalstellen).

4.8 Unterstützung von Softwarelösungen

4.8.1 Überblick

Der nächste Schritt ist die Unterstützung des Interaktionsframeworks durch Softwarelösungen. Das Ziel ist

- die Unterstützung bei der Bearbeitung eines Interaktionsframeworks,
- die Sicherstellung der Vollständigkeit und Gültigkeit eines Interaktionsframeworks,
- die Unterstützung der Übertragbarkeit eines Interaktionsframeworks,
- die Unterstützung des Betriebs von Informationssystemen,
- die Unterstützung der Interoperabilität der Kommunikation.

Bei der Unterstützung von Softwarelösungen werden zwei Ebenen identifiziert. Die erste Ebene betrifft das Interaktionsframework. Die zweite Ebene betrifft die eigentliche Kommunikation, die auf dem Interaktionsframework basiert. Dieser Teil von ISO 29481 gilt für beide Ebenen.

In Bild 6 wird übersichtlich dargestellt, wie die Softwarelösungen unterstützt werden. In den folgenden Abschnitten wird dies erläutert.

4.8.2 Unterstützung des Interaktionsframeworks

Zur Unterstützung der Übertragbarkeit eines Interaktionsframeworks sollte deutlich gemacht werden, welchen Regeln ein Interaktionsframework folgen soll. Diese Regeln müssen in einem Interaktionsframeworkschema enthalten sein, das als XSD-Schemadatei gespeichert wird. Ein Interaktionsframework umfasst Klasseninstanzen, die in diesem Schema definiert werden und in einer XML-Datei gespeichert werden.

BEISPIEL Mit dem Interaktionsframework wird festgelegt, dass die Definition von Attributen (SimpleElementType) und Einschränkungen von Attributen (UserDefinedType) in einem Interaktionsframework enthalten sein dürfen.

In Kapitel 5 werden das Interaktionsframework-Schema und die verfügbaren Klassen beschrieben.

Jedes Interaktionsframework muss die Anforderungen des Interaktionsframework-Schemas erfüllen.

Das Interaktionsframework-Schema sollte verwendet werden, um die erstellten Frameworks zu validieren.

4.8.3 Promotor

Sobald ein gültiges Interaktionsframework vorhanden ist, kann sie mithilfe eines geeigneten Informationssystems interpretiert werden. Dieses System kann dann die Kommunikationen entsprechend der im Interaktionsframework festgelegten Optionen unterstützen. Weiterhin ist es wünschenswert, dass die empfangenen und versendeten Nachrichten validiert werden können, was mithilfe des Interaktionsschemas möglich ist.

Das Interaktionsschema wird mit einem generischen Algorithmus erstellt, der als „Promotor“ bezeichnet wird. Der Promotor weist XML-Instanzen XSD-Klassen zu. Die Inputs sind

- Interaktionsframework (XML),
- Interaktionsframework-Schema (XSD),
- Vorlagensammlung (XSD), die mehrere Vorlagen enthält, die nicht in einem Interaktionsframework verwendet werden, jedoch für jedes Interaktionsschema gültig sind, z. B. für die Kopfzeile der Nachricht.

Der Output ist ein als XSD-Vorlagensammlung gespeichertes Interaktionsschema.

BEISPIEL Der Promotor nimmt Informationen aus dem Interaktionsframework, um das Attribut „CostEstimation“ einzuschließen, das als Pflichtelement für eine bestimmte Nachricht verwendet werden soll, und erstellt ein Interaktionsschema, mit dem die Nachricht mit dem Attribut „CostEstimation“ definiert wird.

In Anhang B wird die Vorlagen-XSD-Datei beschrieben.

Anhang D enthält eine Beschreibung des Promotors.

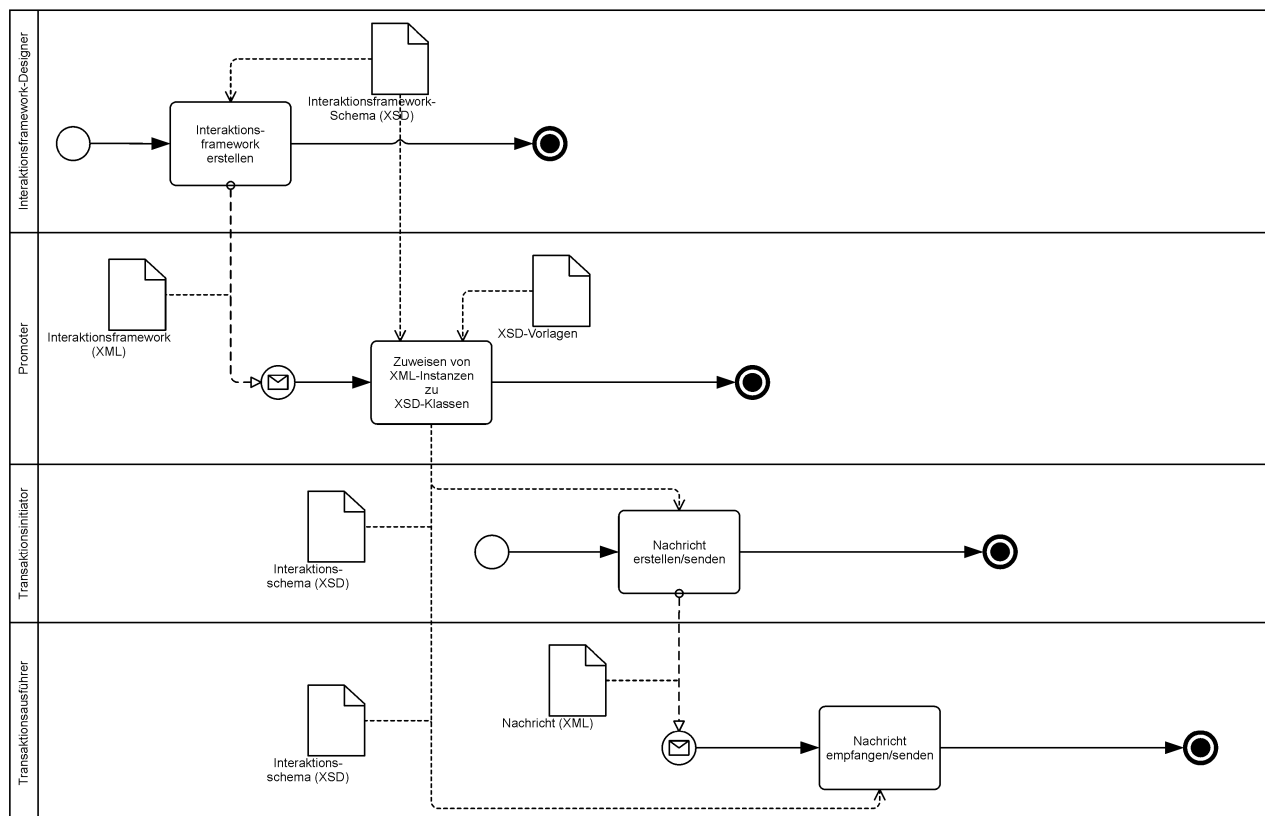


Bild 6 — Unterstützung von Softwarelösungen

4.8.4 Unterstützung der Kommunikation

Jedes Informationssystem, das nach der Definition des Interaktionsframeworks Teil der Kommunikation ist, sollte auf Basis des entsprechenden Interaktionsframeworks und des entsprechenden Interaktionsschemas betrieben werden. Jede Nachricht, die versendet oder empfangen wird, sollte entsprechend des Interaktionsschemas gültig sein.

4.8.5 Technische Implementierung der Kommunikation

Um sicherzustellen, dass die Nachrichten mit Anhängen im technischen Sinne zwischen den Informationssystemen ausgetauscht werden können, sind Implementierungsrichtlinien erforderlich. Zu den darin enthaltenen Themen gehören

- Kommunikationsprotokoll,
- Kommunikationsarchitektur/Server,
- Verschlüsselung,
- SOAP-Funktionsaufrufe.

Implementierungsrichtlinien sind nicht Teil dieser Norm ISO 29481.

5 Format eines Interaktionsframeworks

5.1 Einleitung

In Abschnitt 4 wird beschrieben, dass jedes Interaktionsframework dem Interaktionsframework-Schema entsprechen muss, um Softwarelösungen unterstützen zu können. Dieser Abschnitt wurde hinzugefügt, um das Format eines Interaktionsframeworks mithilfe einer Beschreibung des Interaktionsframework-Schemas zu definieren.

5.2 bietet eine Übersicht über die Informationsklassen, die in einem Interaktionsframework vorkommen können und im Interaktionsframework-Schema definiert werden. Da ein Interaktionsframework im Format XML vorliegt, wird das Wort „Typ“ anstelle von „Klasse“ verwendet. Anhang A enthält eine vollständige Beschreibung eines Interaktionsframework-Schemas. Anhang C enthält ein Beispiel für eine Instanz eines Interaktionsframeworks.

5.2 Informationstypen im Interaktionsframework-Schema

5.2.1 Einleitung

Ein Schema enthält viele Klassen oder Typen. Dieser Abschnitt enthält eine kurze Beschreibung der verfügbaren Typen im Interaktionsframework-Schema. Anhang A enthält eine vollständige Beschreibung in XML eines Interaktionsframework-Schemas. Ein Interaktionsframework muss aus Instanzen dieser Typen erstellt werden und muss über eine Kopfzeile verfügen, die auf das Schema mit den definierten verfügbaren Typen verweist.

5.2.2 AppendixType

„AppendixType“ ist eine Definition zur Festlegung der Struktur von Metadaten-Elementen. Eine Instanz eines „AppendixType“ wird verwendet, um bestimmte Typen von Dateien oder Dokumenten zu definieren, die Teil der versendeten/empfangenen Nachrichten sein können. Die Struktur dieser Elemente in Bezug auf eine Instanz von „AppendixType“ stellt die spezifischen Metadaten dar, die für einen bestimmten Datei- oder Dokumententyp erforderlich sind.

5.2.3 ComplexElementType

„ComplexElementType“ ist eine Ansammlung von „SimpleElementTypes“. Jeder „SimpleElementType“ kommt exakt so oft vor, wie auf ihn verwiesen wird.

5.2.4 ElementCondition

Eine Instanz von „ElementCondition“ beschreibt das Verhalten von Elementwerten in aufeinanderfolgenden Nachrichten. Wenn beispielsweise eine Instanz des Typs „ElementCondition“ mit dem Wert „FIXED“ erstellt wird, weist sie darauf hin, dass Elemente in aufeinanderfolgenden Nachrichten kopiert werden müssen, wenn dasselbe Element vorhanden ist und der Wert nicht geändert werden kann. Eine „ElementCondition“ kann sich auf unterschiedliche Ebenen in einem Framework beziehen. Sie kann sich direkt auf ein „SimpleElement“ beziehen. Jedoch ist es auch möglich, dass sie eine „ElementCondition“ auf ein „ComplexElement“ oder einen „MessageInTransactionType“ verweist. In diesem Fall ist die „ElementCondition“ für alle Elemente gültig, die Teil der Elementenstruktur/-ansammlung der entsprechenden Typen ist.

5.2.5 GroupType

Ein „GroupType“ ermöglicht die Erstellung mehrerer Instanzen einer Gruppe mit spezifischen Inhalten. Der „GroupType“ kann zur Kategorisierung von Nachrichten innerhalb einer Transaktion oder von an Nachrichten angehängten Dokumenten innerhalb einer Transaktion verwendet werden.

5.2.6 MessageType

Der „MessageType“ wird zur Definition der Inhalte einer Nachricht verwendet. Die Elemente, die Teil einer Nachricht sind, werden wiederum in eine oder mehrere Instanzen eines „ComplexElementType“ gruppiert.

5.2.7 MessageInTransactionType

Der „MessageInTransactionType (MITT)“ ist eine Definition, die dazu verwendet wird, einen „MessageType“ auf einen „TransactionType“ zu beziehen. Einfach ausgedrückt: Derselbe „MessageType“ kann mehr als einmal in einem bestimmten „TransactionType“ vorkommen und umgekehrt. Es ist möglich, mehr als eine derselben Instanz eines „MessageType“ auf eine Instanz eines „TransactionType“ und eine Instanz eines „MessageType“ auf mehrere Instanzen eines „TransactionType“ zu beziehen. Darüber hinaus ermöglicht ein MITT die Möglichkeit, die Richtung von Ausführer zu Initiator zu verändern. Eine weitere Option regelt, ob der Nachrichtenfluss durch offene sekundäre Transaktionen blockiert wird oder nicht.

5.2.8 OrganisationType

Definition einer bestimmten Gruppe an Organisationen. Üblicherweise ist mindestens eine Instanz im Framework mit dem Zweck vorhanden, die Elementenstruktur einer Organisation zu definieren.

5.2.9 PersonType

Definition eines Personentyps. Üblicherweise ist mindestens eine Instanz im Framework mit dem Zweck vorhanden, die Elementenstruktur einer Person zu definieren. Der „PersonType“ kann zur Kategorisierung von Personengruppen verwendet werden, die sich auf eine Rolle beziehen.

5.2.10 ProjectType

Definition eines Projekttyps. Üblicherweise ist mindestens eine Instanz im Framework mit dem Zweck vorhanden, die Elementenstruktur eines Projekts zu definieren.

5.2.11 RoleType

Definition einer Rolle. Instanzen eines „RoleType“ sind Voraussetzung zur Erstellung eines „TransactionType“ in einem Framework.

5.2.12 SimpleElementType

Der „SimpleElementType“ ist eine Definition, mit der die Eigenschaften beschrieben werden, die innerhalb von Objektstrukturen auftreten können. Der Bezug auf ein Objekt wird immer über einen „ComplexElementType“ hergestellt.

5.2.13 TransactionPhaseType

Definition, die verwendet werden kann, um eine Instanz zu definieren, mit der die Phase beschrieben wird, in der sich eine Transaktion befindet. Beispielsweise kann eine Instanz eines „TransactionPhaseType“ „Ergebnis angefordert“ oder „Wartend“ sein.

5.2.14 TransactionType

Definition einer Transaktion. In der Instanz einer Transaktion werden die Rollen des Initiators und des Ausführers definiert.

5.2.15 UserDefinedType

Der „UserDefinedType“ wird zur Festlegung von Datentypen (z. B. von einem String) und von XSD-Einschränkungen verwendet. Beispielsweise kann mit einer Instanz eines „UserDefinedType“ die Mindestlänge eines Strings definiert werden.

In Bild 7 wird das Modell des Interaktionsframework-Schemas einschließlich aller Referenzen dargestellt.

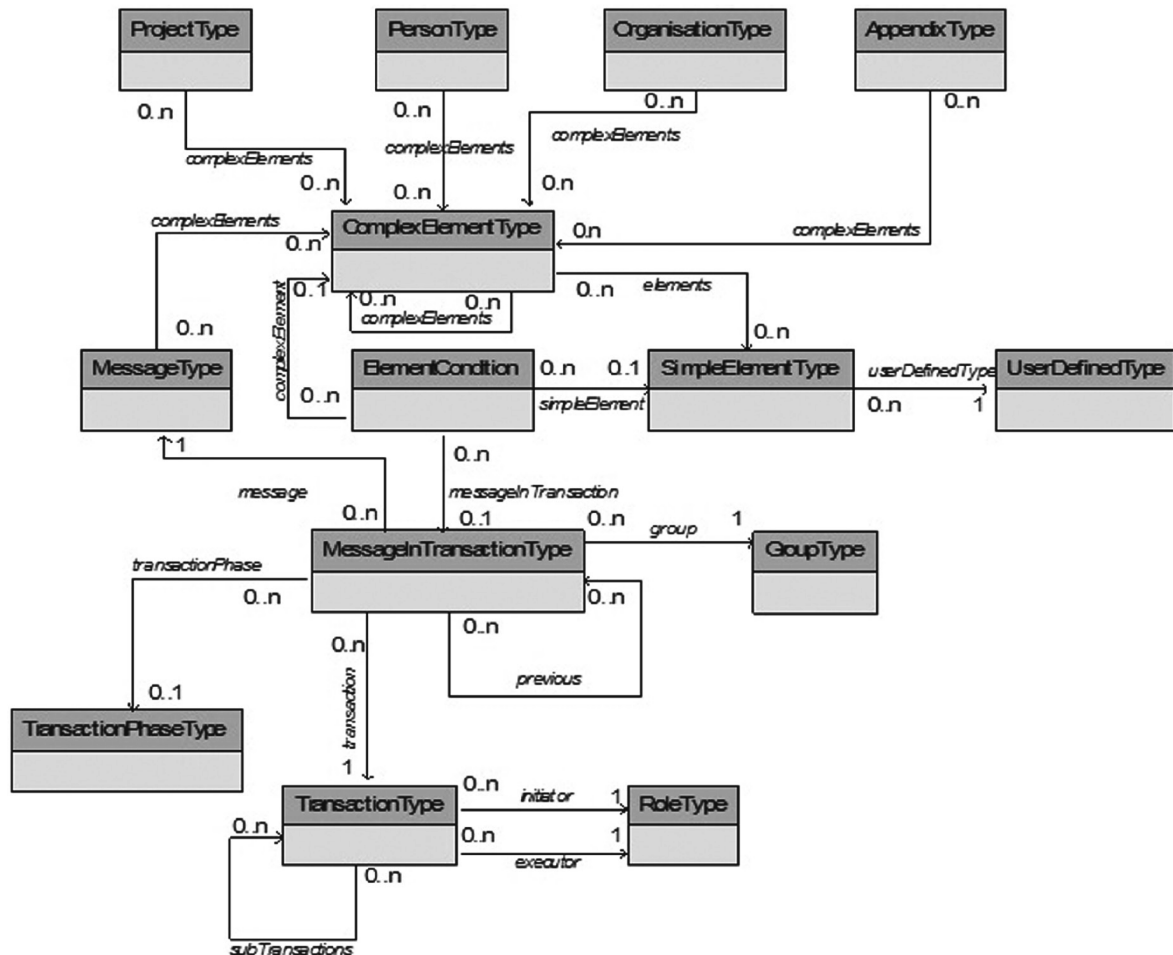


Bild 7 — Typen und Referenzen des Interaktionsframework-Schemas

Anhang A (normativ)

Interaction framework schema definition

A.1 Introduction

An interaction framework must comply with the rules that are included in an interaction framework schema. In this annex, the interaction framework schema definition is given in EXPRESS format and in XSD format. Also, descriptions are given of Element types, attributes, elements, and references.

All objects in an interaction framework require a start date and end date. This requirement is to enable time constraints to the validity of a certain object.

A.2 Interaction framework schema definition (EXPRESS format)

SCHEMA ISO 29481_Part_2A;

ENTITY ProjectType; — The definition of a specific group of projects. Generally, only one instance will be present in an interaction framework defining the structure of elements that each instance of project should expose.

namespace: STRING;

description: STRING;

startDate: DATETIME;

endDate: DATETIME;

state: STRING;

dateLaMu: DATETIME;

userLaMu: STRING;

language: OPTIONAL STRING;

category: OPTIONAL STRING;

helpInfo: OPTIONAL STRING;

code: OPTIONAL STRING;

complexElements: OPTIONAL SET [0:?] OF ComplexElementType;

END_ENTITY;

ENTITY PersonType; — The definition of a specific group of persons. Generally, only one instance will be present in an interaction framework defining the structure of elements that each instance of person should expose.

description: STRING;

startDate: DATETIME;

endDate: DATETIME;

state: STRING;

dateLaMu: DATETIME;

userLaMu: STRING;

language: OPTIONAL STRING;
category: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
code: OPTIONAL STRING;
complexElements: OPTIONAL SET [0:?] OF ComplexElementType;
END_ENTITY;

ENTITY OrganisationType; — The definition of a specific group of organizations. Generally, only one instance will be present in an interaction framework defining the structure of elements that each instance of organization should expose.

description: STRING;
startDate: DATETIME;
endDate: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
language: OPTIONAL STRING;
category: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
code: OPTIONAL STRING;
complexElements: OPTIONAL SET [0:?] OF ComplexElementType;
END_ENTITY;

ENTITY AppendixType; — An AppendixType contains the definition of an appendix. Which data items should be recorded with an appendix can be specified in the complex element section.

description: STRING;
startDate: DATETIME;
endDate: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
language: OPTIONAL STRING;
category: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
code: OPTIONAL STRING;
complexElements: OPTIONAL SET [0:?] OF ComplexElementType;
END_ENTITY;

ENTITY ComplexElementType; — A ComplexElementType contains a set of SimpleElementTypes. Each stated SimpleElementType occurs exactly the number of times it is mentioned.

description: STRING;
startDate: DATETIME;

endDate: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
language: OPTIONAL STRING;
category: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
complexElements: OPTIONAL SET [0:?] OF ComplexElementType;
simpleElements: OPTIONAL SET [0:?] OF SimpleElementType;
END_ENTITY;

ENTITY MessageType; — The definition of a type of message (MessageType), specifying the structure of the message and which set of SimpleElementType's (via ComplexElementType's) may accompany.

description: STRING;
startDate: DATETIME;
endDate: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
language: OPTIONAL STRING;
category: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
code: OPTIONAL STRING;
complexElements: OPTIONAL SET [0:?] OF ComplexElementType;
appendixTypes: OPTIONAL SET [1:?] OF AppendixType;
END_ENTITY;

ENTITY ElementCondition; — The condition of a SimpleElementType as used within a specific MessageType.

description: STRING;
condition: STRING;
helpInfo: OPTIONAL STRING;
complexElement: OPTIONAL ComplexElementType;
simpleElement: OPTIONAL SimpleElementType;
messageInTransaction: OPTIONAL MessageInTransactionType;
END_ENTITY;

ENTITY SimpleElementType; — The definition of a simple element type (SimpleElementType). This element type specifies a property which may occur within various object structures, for example in MessageType (see also AppendixType, ProjectType, PersonType, and OrganisationType). A SimpleElementType is always embedded in a ComplexElementType.

description: STRING;

interfaceType: STRING;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
language: OPTIONAL STRING;
category: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
valueList: OPTIONAL STRING;
userDefinedType: UserDefinedType;
END_ENTITY;

ENTITY UserDefinedType; — A specification of a data type (to be used in a SimpleElementType). This data type encapsulates fill-in areas in the final message, for example a Dutch zip code starts always with four digits and then two characters.

description: STRING;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
baseType: STRING;
xsdRestriction: OPTIONAL STRING;
language: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
END_ENTITY;

ENTITY MessageInTransactionType; — The occurrence of a MessageType within a TransactionType related to a certain group type (GroupType).

requiredNotify: INTEGER;
dateLaMu: DATETIME;
userLaMu: STRING;
received: BOOLEAN;
send: BOOLEAN;
state: STRING;
initiatorToExecutor: OPTIONAL BOOLEAN;
openSecondaryTransactionsAllowed: OPTIONAL BOOLEAN;
firstMessage: OPTIONAL BOOLEAN;
message: MessageType;
previous: OPTIONAL SET [0:?] OF MessageInTransactionType;
transaction: TransactionType;
transactionPhase: OPTIONAL TransactionPhaseType;
group: GroupType;

appendixTypes: OPTIONAL SET [1:?] OF AppendixType;

END_ENTITY;

ENTITY TransactionPhaseType; — The definition of a phase related to a transaction. Examples are 'assignment accepted' or 'result part received'.

description: STRING;

startDate: DATETIME;

endDate: DATETIME;

state: STRING;

dateLaMu: DATETIME;

userLaMu: STRING;

language: OPTIONAL STRING;

category: OPTIONAL STRING;

helpInfo: OPTIONAL STRING;

code: OPTIONAL STRING;

END_ENTITY;

ENTITY TransactionType; — The definition of a type of transaction. A transaction type may reference other transaction types. A transaction will be initiated by a person belonging to an organization fulfilling a certain role. At this level, the role type of the initiator should be stated (the promoted schema will enforce this). The same holds for the executor.

description: STRING;

startDate: DATETIME;

endDate: DATETIME;

state: STRING;

dateLaMu: DATETIME;

userLaMu: STRING;

language: OPTIONAL STRING;

category: OPTIONAL STRING;

helpInfo: OPTIONAL STRING;

code: OPTIONAL STRING;

result: OPTIONAL STRING;

subTransactions: OPTIONAL SET [1:?] OF TransactionType;

initiator: RoleType;

executor: RoleType;

appendixTypes: OPTIONAL SET [1:?] OF AppendixType;

END_ENTITY;

ENTITY RoleType; — The definition of a specific role type.

description: STRING;

startDate: DATETIME;

endDate: DATETIME;

```

state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
language: OPTIONAL STRING;
category: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
code: OPTIONAL STRING;
responsibilityScope: OPTIONAL STRING;
responsibilityTask: OPTIONAL STRING;
responsibilitySupportTask: OPTIONAL STRING;
responsibilityFeedback: OPTIONAL STRING;
END_ENTITY;

ENTITY GroupType; — The definition of a group to store appendices sent with a message within a
transaction.
description: STRING;
startDate: DATETIME;
endDate: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
language: OPTIONAL STRING;
category: OPTIONAL STRING;
helpInfo: OPTIONAL STRING;
END_ENTITY;
END_SCHEMA;

```

A.3 Interaction framework schema definition (XSD format)

```

<?xml version="1.0" encoding = "UTF-8"? >
<xsd:schema targetNamespace="http://www.ISO 29481_Part_2A.com/schemas"
xmlns:iso29481p2a = "http://www.ISO 29481_Part_2A.com/schemas"
xmlns:xsd = "http://www.w3.org/2001/XMLSchema" elementFormDefault = "qualified" >
  <!-- root element declaration (for SCHEMA definitions) -->
  <xsd:element name="ISO 29481_Part_2A" >
    <xsd:complexType>
      <xsd:choice maxOccurs="unbounded" >
        <xsd:element ref="iso29481p2a:AppendixType"/ >
        <xsd:element ref="iso29481p2a:ComplexElementType"/ >
        <xsd:element ref="iso29481p2a:ElementCondition"/ >
        <xsd:element ref="iso29481p2a:GroupType"/ >
        <xsd:element ref="iso29481p2a:MessageInTransactionType"/ >
        <xsd:element ref="iso29481p2a:MessageType"/ >
        <xsd:element ref="iso29481p2a:OrganisationType"/ >
        <xsd:element ref="iso29481p2a:PersonType"/ >
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>

```

```

        <xsd:element ref="iso29481p2a:ProjectType" />
        <xsd:element ref="iso29481p2a:RoleType" />
        <xsd:element ref="iso29481p2a:SimpleElementType" />
        <xsd:element ref="iso29481p2a:TransactionPhaseType" />
        <xsd:element ref="iso29481p2a:TransactionType" />
        <xsd:element ref="iso29481p2a:UserDefinedType" />
    </xsd:choice>
</xsd:complexType>
</xsd:element>
<!-- element declarations (for ENTITY definitions) -->
<xsd:element name="AppendixType" type="iso29481p2a:AppendixTypeType">
    <xsd:annotation>
        <xsd:documentation>An AppendixType contains the definition of an appendix. Which data
items should be recorded with an appendix can be specified in the complex element
section.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="ComplexElementType" type="iso29481p2a:ComplexElementTypeType">
    <xsd:annotation>
        <xsd:documentation>A ComplexElementType contains a set of SimpleElementTypes. Each
stated SimpleElementType occurs exactly the number of times mentioned.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="ElementCondition" type="iso29481p2a:ElementConditionType">
    <xsd:annotation>
        <xsd:documentation>The condition of a SimpleElementType as used within a specific
MessageType.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="GroupType" type="iso29481p2a:GroupTypeType">
    <xsd:annotation>
        <xsd:documentation>The definition of a group to store appendices sent with a message within
a transaction.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="MessageInTransactionType"
type="iso29481p2a:MessageInTransactionTypeType">
    <xsd:annotation>
        <xsd:documentation>The occurrence of a MessageType within a TransactionType related to a
certain group type (GroupType).</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="MessageType" type="iso29481p2a:MessageTypeType">
    <xsd:annotation>
        <xsd:documentation>The definition of a type of message (MessageType), specifying the
structure of the message and which set of SimpleElementType's (via ComplexElementType's) may
accompany.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="OrganisationType" type="iso29481p2a:OrganisationTypeType">
    <xsd:annotation>
        <xsd:documentation>The definition of a specific group of organisations. Generally only one
instance will be present in a interaction framework defining the structure of elements that each instance of
organisation should expose.</xsd:documentation>
    </xsd:annotation>

```

```

</xsd:element>
<xsd:element name="PersonType" type="iso29481p2a:PersonTypeType" >
  <xsd:annotation>
    <xsd:documentation>The definition of a specific group of persons. Generally only one instance
will be present in a interaction framework defining the structure of elements that each instance of person
should expose.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="ProjectType" type="iso29481p2a:ProjectTypeType" >
  <xsd:annotation>
    <xsd:documentation>The definition of a specific group of projects. Generally only one instance
will be present in a interaction framework defining the structure of elements that each instance of project
should expose.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="RoleType" type="iso29481p2a:RoleTypeType" >
  <xsd:annotation>
    <xsd:documentation>The definition of a specific role type.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="SimpleElementType" type="iso29481p2a:SimpleElementTypeType" >
  <xsd:annotation>
    <xsd:documentation>The definition of a simple element type (SimpleElementType). This
element type specifies a property whic may occur within various object structures like for example in
MessageType (see also AppendixType, ProjectType, PersonType and OrganisationType). A
SimpleElementType is always embedded in a ComplexElementType.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="TransactionPhaseType" type="iso29481p2a:TransactionPhaseTypeType" >
  <xsd:annotation>
    <xsd:documentation>The definition of a phase related to a transaction. Examples are
'assignment accepted' or 'result part received'.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="TransactionType" type="iso29481p2a:TransactionTypeType" >
  <xsd:annotation>
    <xsd:documentation>The definition of a type of transaction. A transaction type may reference
again other transaction types. A transaction will be initiated by a person belonging to an organisation
fulfilling a certain role. At this level the role type of the initiator should be stated (the promoted schema will
enforce this). The same holds for the executor.</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<xsd:element name="UserDefinedType" type="iso29481p2a:UserDefinedTypeType" >
  <xsd:annotation>
    <xsd:documentation>A specification of a data type (to be used in a
SimpleElementType).</xsd:documentation>
  </xsd:annotation>
</xsd:element>
<!-- element reference declarations (for ENTITY definitions) -->
<xsd:element name="AppendixTypeRef" type="iso29481p2a:AppendixTypeTypeRef" / >
<xsd:element name="ComplexElementTypeRef"
type="iso29481p2a:ComplexElementTypeTypeRef" / >
<xsd:element name="ElementConditionRef" type="iso29481p2a:ElementConditionTypeRef" / >
<xsd:element name="GroupTypeRef" type="iso29481p2a:GroupTypeTypeRef" / >

```

```

<xsd:element name="MessageInTransactionTypeRef"
type = "iso29481p2a:MessageInTransactionTypeTypeRef" / >
<xsd:element name="MessageTypeRef" type = "iso29481p2a:MessageTypeTypeRef" / >
<xsd:element name="OrganisationTypeRef" type = "iso29481p2a:OrganisationTypeTypeRef" / >
<xsd:element name="PersonTypeRef" type = "iso29481p2a:PersonTypeTypeRef" / >
<xsd:element name="ProjectTypeRef" type = "iso29481p2a:ProjectTypeTypeRef" / >
<xsd:element name="RoleTypeRef" type = "iso29481p2a:RoleTypeTypeRef" / >
<xsd:element name="SimpleElementTypeRef" type = "iso29481p2a:SimpleElementTypeTypeRef" / >
<xsd:element name="TransactionPhaseTypeRef" type = "iso29481p2a:TransactionPhaseTypeTypeRef" / >
>
<xsd:element name="TransactionTypeRef" type = "iso29481p2a:TransactionTypeTypeRef" / >
<xsd:element name="UserDefinedTypeRef" type = "iso29481p2a:UserDefinedTypeTypeRef" / >
<!-- complex types for element declarations (for ENTITY definitions) -->
<xsd:complexType name="AppendixTypeType" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementType" >
      <xsd:sequence>
        <xsd:element name="description" type = "xsd:string" / >
        <xsd:element name="startDate" type = "xsd:dateTime" / >
        <xsd:element name="endDate" type = "xsd:dateTime" / >
        <xsd:element name="state" type = "xsd:string" / >
        <xsd:element name="dateLaMu" type = "xsd:dateTime" / >
        <xsd:element name="userLaMu" type = "xsd:string" / >
        <xsd:element name="language" type = "xsd:string" minOccurs = "0" / >
        <xsd:element name="category" type = "xsd:string" minOccurs = "0" / >
        <xsd:element name="helpInfo" type = "xsd:string" minOccurs = "0" / >
        <xsd:element name="code" type = "xsd:string" minOccurs = "0" / >
        <xsd:element name="complexElements" minOccurs = "0" / >
        <xsd:complexType>
          <xsd:choice maxOccurs="unbounded" >
            <xsd:element ref="iso29481p2a:ComplexElementType" / >
            <xsd:element ref="iso29481p2a:ComplexElementTypeRef" / >
          </xsd:choice>
        </xsd:complexType>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ComplexElementTypeType" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementType" >
      <xsd:sequence>
        <xsd:element name="description" type = "xsd:string" / >
        <xsd:element name="startDate" type = "xsd:dateTime" / >
        <xsd:element name="endDate" type = "xsd:dateTime" / >
        <xsd:element name="state" type = "xsd:string" / >
        <xsd:element name="dateLaMu" type = "xsd:dateTime" / >
        <xsd:element name="userLaMu" type = "xsd:string" / >
        <xsd:element name="language" type = "xsd:string" minOccurs = "0" / >
        <xsd:element name="category" type = "xsd:string" minOccurs = "0" / >
        <xsd:element name="helpInfo" type = "xsd:string" minOccurs = "0" / >
        <xsd:element name="complexElements" minOccurs = "0" / >
        <xsd:complexType>
          <xsd:choice maxOccurs="unbounded" >

```

```

        <xsd:element ref="iso29481p2a:ComplexElementType"/ >
        <xsd:element ref="iso29481p2a:ComplexElementTypeRef"/ >
    </xsd:choice>
</xsd:complexType>
</xsd:element>
<xsd:element name="simpleElements" minOccurs = "0" >
    <xsd:complexType>
        <xsd:choice maxOccurs="unbounded" >
            <xsd:element ref="iso29481p2a:SimpleElementType"/ >
            <xsd:element ref="iso29481p2a:SimpleElementTypeRef"/ >
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ElementConditionType" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementType" >
            <xsd:sequence>
                <xsd:element name="description" type = "xsd:string"/ >
                <xsd:element name="condition" type = "xsd:string"/ >
                <xsd:element name="helpInfo" type = "xsd:string" minOccurs = "0"/ >
                <xsd:element name="complexElement" minOccurs = "0" >
                    <xsd:complexType>
                        <xsd:choice>
                            <xsd:element ref="iso29481p2a:ComplexElementType"/ >
                            <xsd:element ref="iso29481p2a:ComplexElementTypeRef"/ >
                        </xsd:choice>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="simpleElement" minOccurs = "0" >
                    <xsd:complexType>
                        <xsd:choice>
                            <xsd:element ref="iso29481p2a:SimpleElementType"/ >
                            <xsd:element ref="iso29481p2a:SimpleElementTypeRef"/ >
                        </xsd:choice>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="messageInTransaction" minOccurs = "0" >
                    <xsd:complexType>
                        <xsd:choice>
                            <xsd:element ref="iso29481p2a:MessageInTransactionType"/ >
                            <xsd:element ref="iso29481p2a:MessageInTransactionTypeRef"/ >
                        </xsd:choice>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="GroupTypeType" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementType" >

```