

MODELADO DE LA INFORMACIÓN DE LOS EDIFICIOS

MANUAL DE ENTREGA DE LA INFORMACIÓN

**PARTE 2: MARCO DE TRABAJO PARA LA INTERACCIÓN
(ISO 29481-2:2012)**

**Correspondencia: UNE-EN ISO 29481-1:2017
I.C.S.: 935.240.67; 91.010.01**

PREFACIO

Esta Norma Técnica en su etapa de proyecto, ha sido sometida a un período de discusión pública de sesenta (30) días.

La Norma Técnica ISO DGNTI UNE-EN 29481-2:2012 ha sido oficializada por el Ministerio de Comercio e Industrias mediante la resolución N° _____ del ____ de _____ de _____, y publicada en Gaceta Oficial N° _____ del ____ de _____ de _____.

Esta Norma Técnica DGNTI ISO UNE-EN 29481-2:2017 es equivalente a la Norma ISO 29481-2:2012.

Prólogo de la versión en español

Esta Norma Técnica ISO DGNTI UNE-EN 29481-2:2012 es una adopción idéntica de la Norma Internacional ISO 29481-2:2012, la cual fue traducida por la Asociación Española de Normalización (UNE) y fue aprobada por el Comité Europeo de Normalización (CEN).

0. Introducción

El modelado de la información de edificios proporciona un concepto para describir y mostrar la información requerida en el diseño, construcción y operación de las instalaciones construidas. Puede reunir los diversos conjuntos de la información utilizados en la construcción en un entorno de información común, reduciendo, y a menudo eliminando, la necesidad tener en papel muchos tipos de documentación de los que actualmente se usan.

Un manual de entrega de información (IDM, *Information Delivery Manual*) proporciona una ayuda significativa para obtener el máximo beneficio de un modelo de información de un edificio o construcción (BIM, *Building Information Model*). Si la información requerida está disponible cuando se necesita y la calidad de la información es satisfactoria, el proceso de construcción en sí se mejora mucho. Para que esto ocurra debería haber una comprensión común de los procesos de construcción y de la información que se necesita y resulta de su ejecución.

Esta parte de la Norma ISO 29481 se centra en aspectos del proceso de construcción que se refieren a la gestión y coordinación de las partes involucradas. La coordinación depende de la comunicación, que debería estar bien estructurada, ser inequívoca, explícita y ágil. Como se incide mucho en la coordinación y en la interacción, esta parte de la Norma ISO 29481 proporciona un complemento natural a las normas que se centran en la construcción de modelos, como son las Normas ISO 10303-239 e ISO 16739.

Esta parte de la Norma ISO 29481 establece una metodología y un formato para describir las labores de coordinación entre los actores de un proyecto de construcción. Describe cómo identificar y definir los procesos de coordinación emprendidos y la información requerida para su ejecución. Los marcos de interacción resultantes permiten la estandarización de la interacción en los procesos de construcción a nivel nacional, local y de proyecto. También ofrece un formato para dar soporte a las soluciones proporcionadas por los proveedores de soluciones TIC. Esta parte de la Norma 29481 aúna diferentes sistemas de gestión de procesos para dar soporte a diferentes soluciones TIC. De este modo, proporciona una base para el intercambio y la compartición fiable de información entre los usuarios, para que puedan confiar en que la información que envían o reciben es exacta y suficiente para las actividades de coordinación que necesitan realizar.

El desarrollo de esta parte de la Norma ISO 29481 ha sido impulsado por la necesidad de los usuarios de contar con un intercambio de información fiable. Se basa principalmente en el estándar holandés VISI desarrollado en 2003.

Índice

Prefacio.....	2
Prólogo de la versión en español	3
0. Introducción	4
1. Objeto y campo de aplicación	11
2. Normas para consulta	11
3. Términos, definiciones y símbolos.....	11
3.1 Términos y definiciones.....	11
3.1.1 Términos generales.....	12
3.1.2 Términos relacionados con activos y proyectos.....	12
3.1.3 Términos relacionados con gestión de la información	12
3.2 Símbolos.....	13
4. Gestión de la información durante la fase de desarrollo de ls activos.....	13
5. Proceso de gestión de la información durante la fase de desarrollo de los activos.....	14
5.1 Proceso de gestión de la información – Evaluación de necesidades.....	14
5.1.1 Designar a los responsables de la función de gestión de la información.....	14
5.1.2 Establecer los requisitos de información del proyecto.....	15
5.1.3 Establecer los hitos de entrega de la información del proyecto	15
5.1.4 Establecer la norma de información del proyecto.....	16
5.1.5 Establecer los métodos y procedimientos para la producción de información del proyecto.....	16
5.1.6 Establecer la información de referencia del proyecto y los recursos compartidos	17
5.1.7 Establecer el entorno común de datos del proyecto.....	17
5.1.8 Establecer el protocolo de intercambio de información del proyecto	18
5.1.9 Actividades de la evaluación de necesidades	19
5.2 Proceso de gestión de la información – Petición de ofertas	19
5.2.1 Establecer los requisitos de intercambio de información de la parte contratante	19
5.2.2 Reunir la información de referencia y los recursos compartidos.....	21
5.2.3 Establecer los requisitos de presentación de ofertas y los criterios de evaluación	21
5.2.4 Recopilar la información relativa a la licitación	21
5.2.5 Actividades de la petición de ofertas	22

5.3	Proceso de gestión de la información – presentación de ofertas	22
5.3.1	Designación de los responsables de la función de gestión de la información...	22
5.3.2	Establecer el plan de ejecución del BIM del equipo de desarrollo (antes de la contratación)	23
5.3.3	Evaluación de las aptitudes y capacidades del equipo de trabajo	24
5.3.4	Establecer las aptitudes y capacidades del equipo de desarrollo	25
5.3.5	Establecer plan de movilización del equipo de desarrollo	25
5.3.6	Establecer cuadro de riesgos del equipo de desarrollo	26
5.3.7	Recopilar la información de la oferta del equipo de desarrollo	27
5.3.8	Actividades de la presentación de ofertas	27
5.4	Proceso de gestión de la información – contratación	28
5.4.1	Confirmar el plan de ejecución del BIM del equipo de desarrollo	28
5.4.2	Establecer la matriz detallada de responsabilidades del equipo de desarrollo ...	28
5.4.3	Establecer los requisitos de intercambio de información de la parte contratada principal	29
5.4.4	Establecer el/los programa/s de desarrollo de información de una tarea	30
5.4.5	Establecer programa general de desarrollo de la información	30
5.4.6	Completar los documentos de contratación de la parte contratada principal	31
5.4.7	Completar los documentos de contratación de la parte contratada	31
5.4.8	Actividades de contratación	32
5.5	Proceso de gestión de la información – Movilización	32
5.5.1	Movilizar recursos	32
5.5.2	Movilizar tecnología de la información	33
5.5.3	Probar los métodos y procedimientos de producción de información del proyecto	33
5.5.4	Actividades para la movilización	34
5.6	Proceso de gestión de la información – Producción colaborativa de la información	34
5.6.1	Comprobar la disponibilidad de la información de referencia y de los recursos compartidos	34
5.6.2	Producir información	34
5.6.3	Realizar un control de calidad	35
5.6.4	Revisar y aprobar el intercambio de información	36
5.6.5	Revisar el modelo de información	36
5.6.6	Actividades para la producción colaborativa de la información	37

5.7	Proceso de gestión de la información – Entrega del modelo de información	37
5.7.1	Presentar a la parte contratada principal el modelo de información para su autorización	37
5.7.2	Revisar y autorizar el modelo de información.....	37
5.7.3	Presentar a la parte contratante el modelo de información para su aceptación	38
5.7.4	Revisar y autorizar el modelo de información.....	38
5.7.5	Actividades para la entrega del modelo de información	39
5.8	Proceso de gestión de la información – Fin de la fase de desarrollo	39
5.8.1	Archivar el modelo de información del proyecto.....	39
5.8.2	Recoger las lecciones aprendidas para futuros proyectos	40
5.8.3	Actividades del fin de la fase de desarrollo	40
	Anexo A (Informativo) Ilustraciones de estrategias de federación y estructuras de distribución de contenedores de información	41
	Bibliografía.....	43

MODELADO DE LA INFORMACIÓN DE LOS EDIFICIOS
MANUAL DE ENTREGA DE LA INFORMACIÓN
PARTE 2: MARCO DE TRABAJO PARA LA INTERACCIÓN

1. OBJETO Y CAMPO DE APLICACIÓN

Esta parte de la Norma ISO 29481 especifica una metodología y un formato para describir las “acciones de coordinación” entre los agentes de un proyecto de construcción durante todas las etapas del ciclo de vida del bien.

Por lo tanto, especifica:

- Una metodología que describe un marco de interacción,
- Una manera adecuada de asignar responsabilidades e interacciones, que proporciona un contexto del proceso para el flujo de la información,
- Un formato en el que se debería especificar un marco de interacción.

Esta parte de Norma ISO 29481 pretende facilitar la interoperabilidad entre las aplicaciones de *software* utilizadas en el proceso de construcción, promover la colaboración digital entre los agentes en el proceso de construcción de edificios y proporcionar una base para un intercambio de información precisa, fiable, repetible y de alta calidad.

2. NORMAS PARA CONSULTA

Los documentos indicados a continuación, en su totalidad o en partes, son normas para consulta indispensables para la aplicación de este documento. Para las referencias con fecha, solo se aplica la edición citada. Para las referencias sin fecha se aplica la última edición (incluida cualquier modificación de esta).

ISO 29481-1, *Modelado de la información de los edificios. Manual de entrega de la información. Parte 1: Metodología y formato.*

3. TÉRMINOS y DEFINICIONES

Para los fines de este documento, se aplican los términos y definiciones siguientes:

3.1 IDM, Manual de entrega de la información:

Documentación que recoge el proceso de negocio y da especificaciones detalladas de la información que cualquier agente del proyecto podría necesitar proporcionar en cada momento particular dentro de un proyecto.

3.2 Marco de interacción:

Descripción formal de los elementos de interacción, incluyendo la definición de atribuciones, transacciones, mensajes en la transacción y datos en los mensajes.

3.3 Esquema del marco de interacción

Descripción formal de las atribuciones con las que han de cumplir los marcos de interacción

3.4 Esquema de interacción:

Descripción formal de las atribuciones con las que se han de completar los mensajes enviados y recibidos.

3.5 Promotor:

Algoritmo que genera un esquema de interacción desde un marco de interacción, un esquema del marco de interacción y ficheros de plantillas como entradas.

3.6 Archivo de plantillas:

Archivo que contiene una serie de plantillas, independiente del marco de interacción, para generar un esquema de interacción.

3.7 VISI:

Acrónimo del estándar holandés para la comunicación entre socios en los proyectos de construcción.

Nota 1: VISI viene de "*Voorwaarden scheppen voor Invoeren Standaardisatie ICT in de Infrastructuursector*", lo que se traduce como "Creación de condiciones para la implementación de la estandarización de TICs para la industria de la construcción.

4. Principios de normalización

4.1 Generalidades

Este apartado se incluye para destacar y ayudar a explicar los conceptos esenciales en los que se basa esta parte de la Norma 29481.

4.2 BIM y IDM

El modelado de la información de los edificios aúna los diversos conjuntos de información utilizados en la construcción en un entorno de información común. Para que esto suceda, debería haber una comprensión común de los procesos de construcción y la información que se necesita para y desde su ejecución.

La Norma ISO 29481 es un estándar que establece un método para el desarrollo de un Manual de Entrega de la Información.

La metodología de IDM dada en la Norma ISO 29481-1 debe utilizarse para todas las referencias para el desarrollo y uso de IDM.

4.3 Componentes del IDM

La metodología y los componentes del IDM son descritos en la Norma ISO 29481-1. En esta parte se aporta una ilustración que muestra esquemáticamente cuáles son los distintos componentes del IDM y cómo están relacionados entre sí.

Dentro del IDM hay dos perspectivas. Puede verse como requisitos de usuarios o como soluciones técnicas. Dentro de las dos perspectivas, hay una serie de zonas que caracterizan a los distintos componentes del IDM (véase la figura 1).

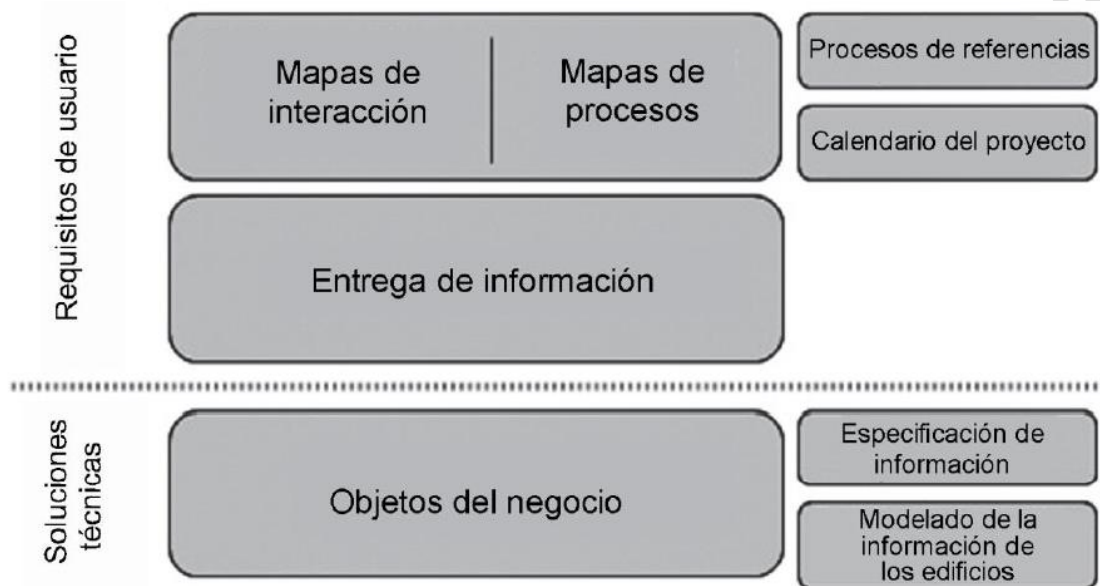


Figura 1 – Zonas del IDM

Dentro de la perspectiva de los requisitos de usuario las zonas son:

- Mapa de interacción, describiendo las atribuciones de cada agente y la relación entre ellos,
- Mapas de procesos, haciendo una descripción de aquellos procesos en los que ocurre el intercambio de información
- Entrega de información, describiendo las necesidades del intercambio de la información.
- Procesos de referencias, (descripciones de intercambio almacenadas),
- El calendario del proyecto, (ocurrencias de sucesos en el contexto de un proyecto),

La perspectiva de soluciones técnicas incluye

- Los objetos del negocio incluyendo los requisitos de intercambio del modelo,
- Las especificaciones de información, describiendo el esquema en el que se basa el intercambio de información,
- El modelado de la información de los edificios.

Esta parte de la Norma ISO 29481 se centra en el mapa de interacción y se basa en los principios generales de la comunicación empresarial.

4.4 Principios básicos de la comunicación empresarial

Una vez que un cliente ha pedido la entrega un producto o un servicio, se pone en marcha una cadena de actividades en funcionamiento, cuyo efecto combinado es proporcionar el producto o servicio. Esta cadena de actividades se denomina proceso de negocios. Más específicamente, hablamos aquí de un proceso de negocio primario porque se inicia externamente.

Parte del proceso de negocios es la comunicación entre las partes involucradas. Esta parte de la norma ISO 29481 se centra en la comunicación que se relaciona con la entrega de un resultado (comunicación performativa). La iniciación y ejecución de una solicitud es a través de acciones comunicativas. En una acción comunicativa, dos partes están siempre involucradas: la persona que realizó la acción y la persona a quien se dirige la acción. El manejo de una solicitud ocurre en un patrón particular llamado transacción.

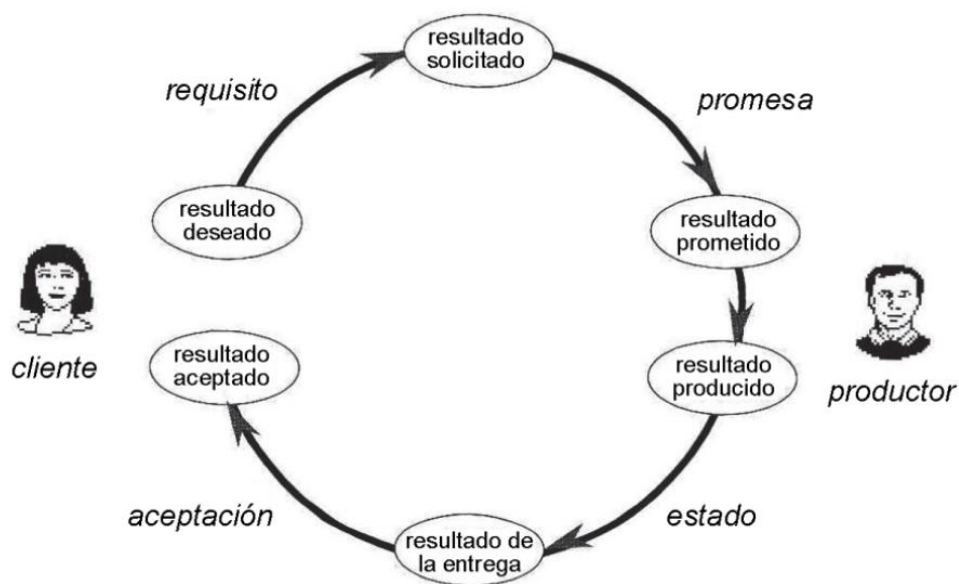


Figura 2 – El patrón de una transacción

En la figura 2, se presenta la forma más simple de este patrón de transacción. Se demuestra que la creación de un nuevo resultado de producción (por ejemplo, el "resultado deseado" es la entrega de un documento) comienza con la solicitud de este resultado por alguien en el papel de cliente a alguien en el papel de productor. Esto lleva el proceso al estado "resultado solicitado". El productor responde a la solicitud prometiendo producir el resultado deseado, lo que lleva el proceso al estado "resultado prometido". Esto representa una tarea para el productor: tiene que cumplir con la promesa al preparar realmente el documento y decidir entregárselo. En el acto de

entrega del documento al cliente, el productor afirma que ha cumplido con su promesa. El cliente responde a este estado aceptando el resultado tal como se produce. Esta acción completa la transacción.

En la ejecución del proceso de un negocio, a menudo participan muchos actores. Su comportamiento depende de sus atribuciones en el proceso. Los roles/actores hacen negocios con otros roles/actores ejecutando transacciones. Una representación útil de la interacción entre roles/actores se denomina mapa de interacción.

4.5 Mapa de interacción

Un mapa de interacción debe identificar los tipos de atribuciones de cada actor y los tipos de transacciones relevantes para un proceso determinado. IDM establece una distinción entre un rol del que hace una solicitud, el iniciador y el rol del que da efecto a esa petición, el ejecutor. Una transacción sólo debe tener su iniciador y un solo ejecutor. La figura 3 muestra los componentes del mapa de interacción.

Nota: La notación del mapa de interacción se basa en el modelo de construcción descrito en la publicación del Prof. Jan L.G. Dietz. Esta notación difiere de BPMN (*Business Process Model and Notation*, en español *Modelo y Notación de Procesos de Negocios*) y se utiliza para preparar mapas de forma que sean tan simples como sea posible. Además, proporciona el concepto de “transacción”, que no está disponible en BPMN.

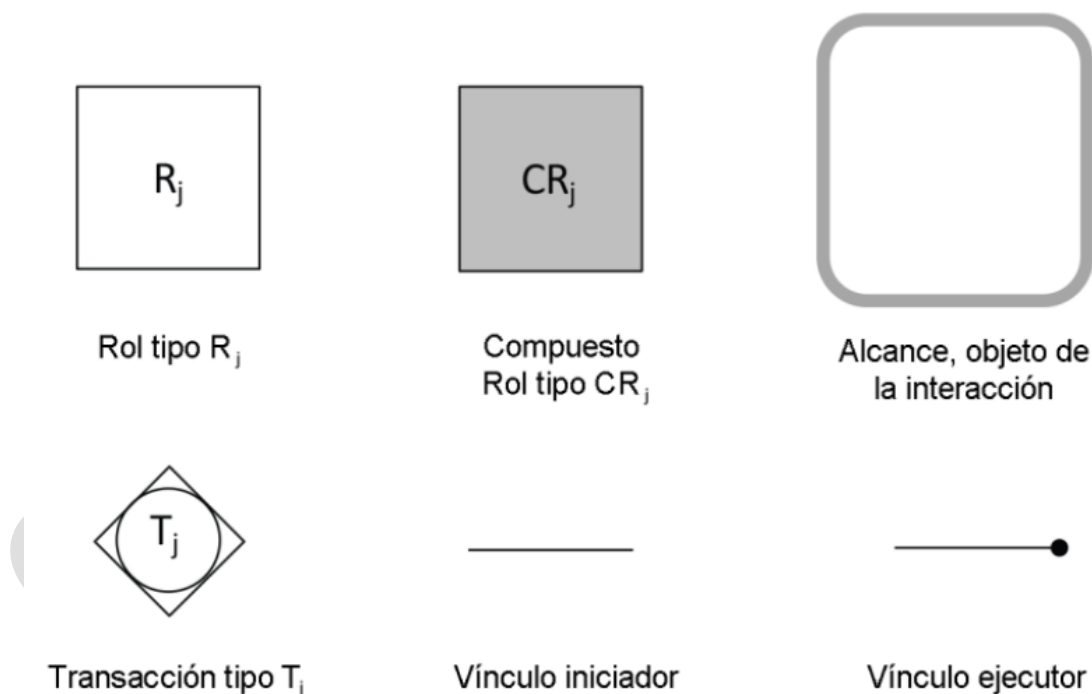


Figura 3 – Componentes de un mapa de interacción

La ventaja del mapa de interacción es que concentra la atención en las relaciones entre los roles o atribuciones de cada actor y, a la vez, esconde tanto la complejidad del

proceso dentro del rol de cada actor, como los detalles de la interacción entre estos roles. El uso de roles abstractos hace que el mapa de interacción sea válido para muchas situaciones diferentes. El mapa de interacción es una valiosa herramienta para analizar y definir elementos esenciales de un proceso de negocio. La figura 4 muestra un ejemplo simplificado de un mapa de interacción de una oficina de diseño.

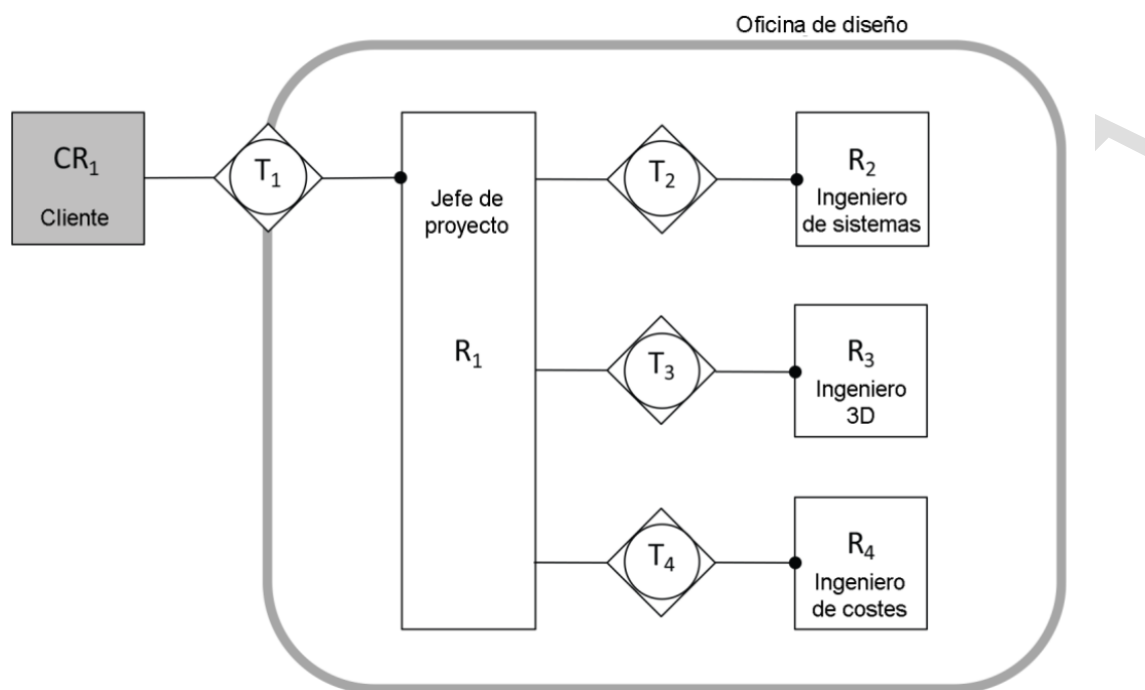


Figura 4 – Ejemplo de un mapa de interacción

En un mapa de interacción deben incluirse todas las transacciones necesarias para el manejo de las contribuciones requeridas de funciones relevantes al BIM. Todos los roles y transacciones dentro del mapa de interacción deben tener una identidad y un nombre únicos. La numeración es arbitraria. El nombre de la función se deriva de la actividad de la actividad principal realizada por cada actor; esto centra la contribución de los roles dentro del BIM. Un papel compuesto es un papel que puede consistir en múltiples roles, pero cuya composición es desconocida o no relevante.

Las interacciones se pueden resumir en una tabla de transacciones.

Tabla 1 – Tabla de transacción de una oficina de diseño simplificada

Resultado de la transacción	Tipo de transacción
Entrega del diseño	T ₁ , Entregar diseño
Entrega del sistema de especificación	T ₂ , Entregar el sistema de especificación
Entrega del modelo 3D	T ₃ , Entregar el modelo 3D
Entrega del presupuesto	T ₄ , Entregar el presupuesto

4.6 Los mensajes en la transacción

Una transacción debe contener un conjunto de mensajes que se intercambian para un propósito particular. La transacción también estipula los papeles participantes, el punto en el ciclo de vida y la secuencia en la que se deberían entregar los mensajes (si procede).

Un ejemplo de una transacción es el manejo de una solicitud para un modelo 3D. Véase la figura 5, que muestra los mensajes en una transacción es el manejo de una solicitud para un modelo 3D. Véase la figura 5, que muestra los mensajes en una transacción como un diagrama de secuencia en una notación UML. La transacción sólo puede iniciarse con R1 Jefe de proyecto con el mensaje 'Solicitud de modelo 3D'. El ingeniero 3D (rol R3) puede responder con un mensaje "Trabajo hecho y pendiente de aprobación". Después de un mensaje "Trabajo aprobado" o "Trabajo no aprobado", se completa la transacción.

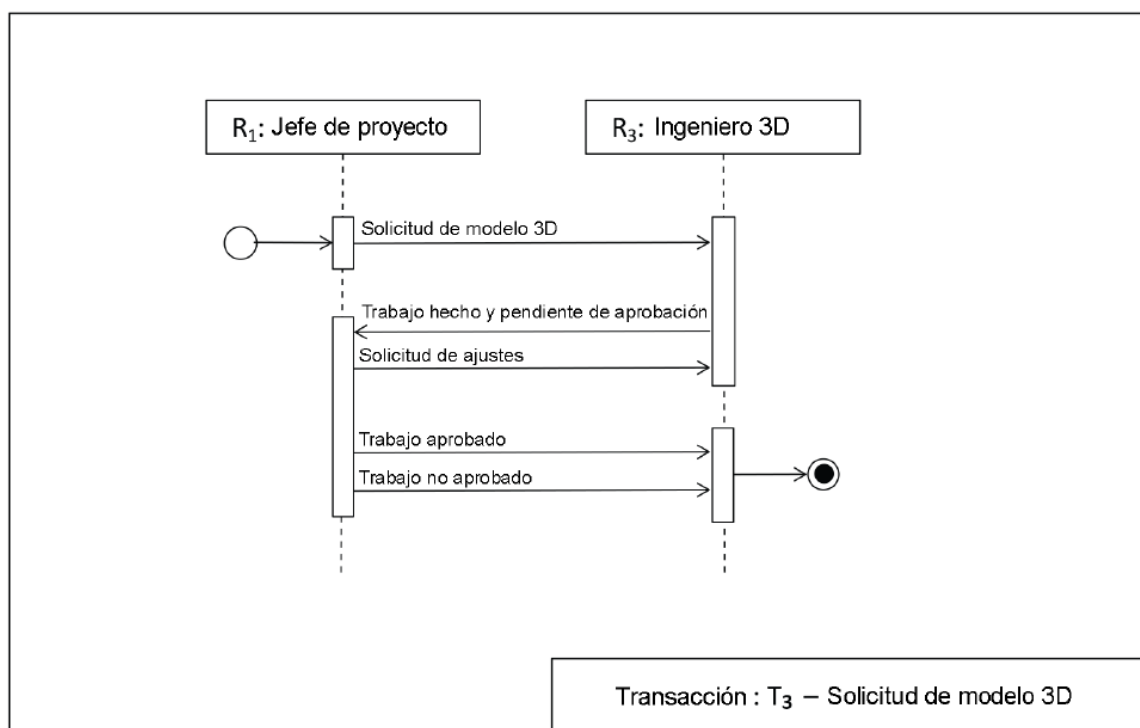


Figura 5 – Ejemplo de mensajes en una transacción

4.7 Marco de interacción

Con el fin de orientar un proceso y la transferencia de información, los elementos de interacción se describirán de manera coherente. Esta descripción coherente se denomina marco de interacción. Un marco de interacción debe incluir:

- Definición de roles,
- Transacciones,

- Mensajes de la transacción,
- El orden de los mensajes de la transacción,
- Los elemento de datos en el mensaje.

Se puede preparar un marco de interacción para un área de aplicación definida y utilizarse como un estándar a nivel internacional, nacional, de organización o de proyecto. Por ejemplo, en los Países Bajos, se desarrolla un marco de interacción a nivel nacional para la finalización de todos los procedimientos contractuales durante la ejecución de los proyectos de construcción. Esta parte de la Norma ISO 29481 puede ser utilizada como una plantilla para organizaciones y proyectos y también permite ser ajustada a las necesidades específicas de cada caso.

EJEMPLO: Un marco de interacción puede incluir el atributo CostEstimation como un ejemplar de SimpleElementType que se utilizará como elemento obligatorio para un determinado mensaje. También puede incluir una restricción en el formato de atributo CostEstimation (por ejemplo, sólo eutos con dos decimales).

4.8 Apoyo de la solución de software

4.8.1 Visión del conjunto

El siguiente paso es apoyar el marco de interacción con soluciones de *software*. El objetivo es:

- Apoyar la edición de un marco de interacción,
- Garantizar la integridad y validez de un marco de interacción,
- Apoyar la portabilidad de un marco de interacción
- Apoyar el funcionamiento de los sistemas de información
- Apoyar la interoperabilidad de la comunicación

Se pueden identificar dos niveles en lo que respecta al apoyo de soluciones de *software*. El primer nivel se refiere al marco de interacción. El segundo nivel se refiere a la comunicación real en la que se basa el marco de interacción. Esta parte de la Norma ISO 29481 se aplica a ambos niveles.

En la figura 6 se ofrece una descripción general de cómo funcionan las soluciones de *software*. En las secciones siguientes se da una explicación de las figuras.

4.8.2 Apoyo al marco de interacción

A fin de apoyar la portabilidad de un marco de interacción, debería quedar claro qué reglas ha de cumplir un marco de interacción. Estas reglas deben incluirse en un esquema de marco de interacción, que se graba como un archivo de esquema XSD. Un marco de interacción comprende ejemplares de clases definidas en el esquema, y debe registrarse como un archivo XML.

EJEMPLO: El esquema del marco de interacción define que se puede incluir la definición de atributos (SimpleElementType) y las restricciones a los

atributos (UserdefinedType) en un marco de interacción.

El capítulo 5 describe el esquema del marco de interacción y las clases disponibles.

Todos los marcos de interacción deberían cumplir con el esquema del marco de interacción.

Para validar el marco de interacción debería usarse un editor de la estructura del esquema del marco de interacción.

4.8.3 Promotor

Una vez disponible un marco de interacción válido, puede ser interpretado por un sistema de información adecuado. A continuación, este sistema puede soportar las comunicaciones de acuerdo con las opciones establecidas en el marco de interacción. Finalmente, es deseable poder validar los mensajes recibidos y enviados. Esto se hace con el esquema de interacción.

El esquema de interacción se genera con un algoritmo genérico llamado Promotor. El Promotor 'promueve' ejemplares XML en clases XSD. La entrada es:

- Marco de interacción (XML),
- Esquema del marco de interacción (XSD),
- Archivo de plantillas (XSD), que contiene un número de plantillas no descritas en un marco de interacción, pero son válidos para cada esquema de interacción, por ejemplo el encabezado del mensaje.

La salida es un esquema de interacción se salva como un archivo XSD.

EJEMPLO: El 'Promotor' toma información del marco de interacción para incluir el atributo CostEstimation para ser utilizado como un elemento obligatorio para un determinado mensaje y crea un esquema de interacción que define el mensaje con el atributo CostEstimation.

El anexo B describe el archivo XSD de plantillas.

El anexo D da los principios del Promotor.

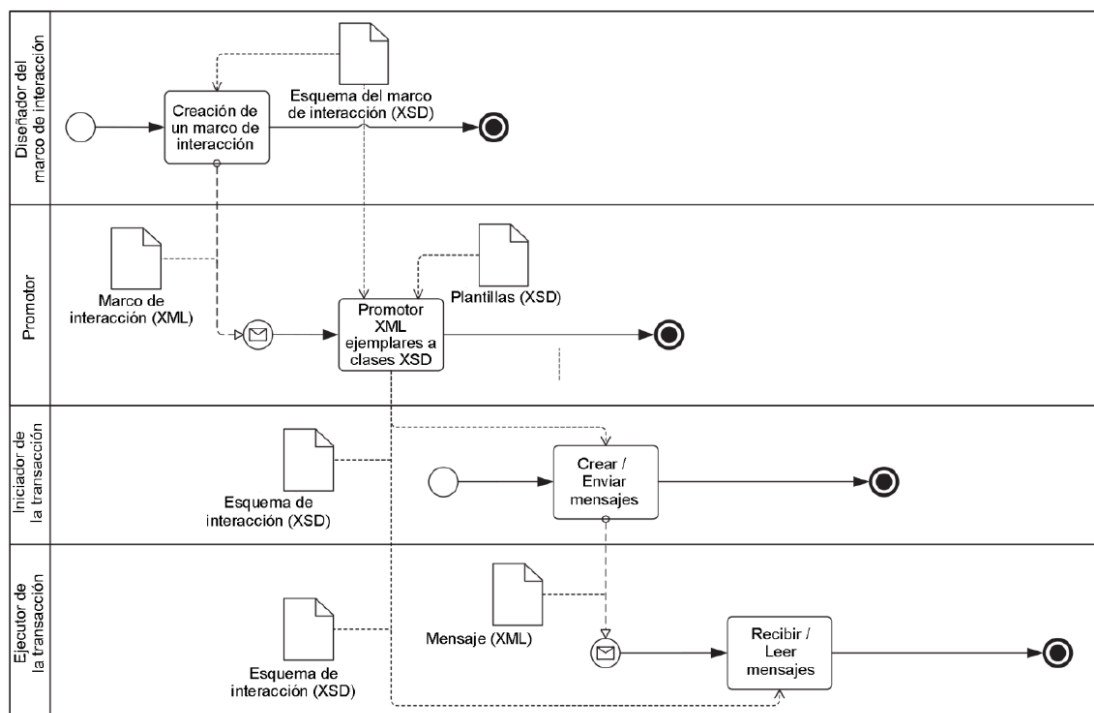


Figura 6 – Validez de las soluciones de software

4.8.4 Apoyo a la comunicación

Todo sistema de información que participe en la comunicación, tal como se define en el marco de interacción, debería operar sobre la base del marco de interacción y del esquema de interacción correspondientes. Cada mensaje enviado o recibido debería ser válido según el esquema de interacción.

4.8.5 Implementación técnica de la comunicación

Con el fin de garantizar que los mensajes con archivos adjuntos en sentido técnico puedan intercambiarse entre sistema de información, se necesitan directrices de implementación. Los temas a cubrir incluyen:

- Protocolo de comunicación,
- Arquitectura/servidores de comunicaciones,
- Cifrado,
- Llamadas de función SOAP.

Las directrices de aplicación están fuera del alcance de esta Norma ISO 29481.

5. Formato de un marco de interacción

5.1 Introducción

El capítulo 4 explica que, en orden, para apoyar las soluciones de software, cada marco

de interacción ha de cumplir con el esquema de la estructura de interacción. Este capítulo se incluye para definir el formato de un marco de interacción a través de una descripción del esquema del marco de interacción.

El apartado 5.2 proporciona una visión general de las clases de información que pueden darse en un marco de interacción y se definen en el esquema del marco de interacción. Dado que un marco de interacción se define en XML, la palabra “tipo” se utiliza en lugar de la clase. El anexo A de la descripción completa del esquema del marco de interacción. El anexo C proporciona un ejemplo de un ejemplar de un marco de interacción.

5.2 Tipos de información en el esquema del marco de interacción

5.2.1 Introducción

Un esquema consta por muchas clases o tipos. En esta sección, se da una breve descripción de los tipos disponibles en el esquema del marco de interacción. El anexo A contiene una descripción completa en XML del esquema del marco de interacción. Debe crearse un marco de interacción a partir de ejemplares de estos tipos disponibles definidos.

5.2.2 AppendixType

El AppendixType es una definición para definir la estructura de los elementos referentes a los metadatos. Un ejemplar de un tipo apéndice se utiliza para definir ciertos tipos de archivos o documentos que pueden formar parte de los mensajes de envío/recepción. La estructura de los elementos relacionados con una instancia de un AppendixType representa los metadatos específicos que se requieren para cierto tipo de archivo o documento.

5.2.3 ComplexElementType

Un ComplexElementType es una colección de SimpleElementTypes. Cada SimpleElementType se da exactamente el número de veces que está relacionado.

5.2.4 ElementCondition

Un ejemplar de un ElementCondition describe el comportamiento de los valores de los elementos en los mensajes sucesivos. Por ejemplo, cuando se crea una instancia del tipo ElementCondition con el valor FIXED, indica que los elementos de los mensajes sucesivos han de copiarse cuando el mismo elemento está disponible y no se puede cambiar el valor. Un ElementCondition puede referirse a diferentes niveles en un marco. Puede estar directamente relacionado con un SimpleElement, pero también es posible relacionar un ElementCondition con un ComplexElement o un MessageInTransactionType. En este caso, el ElementCondition es válido para todos los elementos que forman parte de la estructura/colección de elementos de los tipos relacionados.

5.2.5 GroupType

Un GroupType hace posible crear varios ejemplares de un grupo con su propio contenido específico. El GroupType puede utilizarse para categorizar mensajes dentro de una transacción o documentos relacionados con mensajes dentro de una transacción.

5.2.6 MessageType

MessageType se utiliza para definir el contenido de un mensaje. Los elementos que forman parte de un mensaje son, a su vez, agrupados en uno o más ejemplares de un ComplexElementType.

5.2.7 MessageInTransactionType

MessageInTransactionType (MITT) es una definición que se utiliza para relacionar MessageType con TransactionType. El mismo tipo de mensaje puede darse más de una vez en un tipo de transacción dada y viceversa. Es posible relacionar más de un ejemplar de MessageType a un ejemplar de TransactionType y un ejemplar de MessageType a varios ejemplares de TransactionType. Además, un MITT ofrece una opción para invertir la dirección de ejecutor a iniciador. Otra opción controla si el flujo de mensajes bloqueado por transacciones secundarias abiertas o no.

5.2.8 OrganisationType

Es la definición de un cierto grupo de organizaciones. En común, al menos un ejemplar está disponible en el marco con el objeto de definir la estructura de los elementos de una organización.

5.2.9 PersonType

Es la definición de un tipo de persona. En común, al menos un ejemplar está disponible en el marco de trabajo con el objetivo de definir la estructura de los elementos que definen a una persona. PersonType se puede utilizar para categorizar grupos de personas que están relacionadas con un rol.

5.2.10 ProjectType

Es la definición de un tipo de proyecto. En común, al menos un ejemplar está disponible en el marco con el objeto de definir la estructura de los elementos para definir el proyecto.

5.2.11 RoleType

Es la definición de un rol. Los ejemplares de un RoleType son requisitos previos para crear un TransactionType en un framework.

5.2.12 SimpleElementType

El SimpleElementType es una definición que describe las propiedades que pueden darse dentro de estructuras de objetos. La relación con un objeto es siempre a través de un ComplexElementType.

5.2.13 TransactionPhaseType

Es la definición que se puede utilizar para definir un ejemplar que describe la fase en la que se encuentra una transacción. Por ejemplo, un ejemplar de TransactionPhaseType puede ser “resultado solicitado” o “retenido”.

5.2.14 TransactionType

Es la definición de una transacción. En un ejemplar de una transacción, se definen las funciones del iniciador y del ejecutor.

5.2.15 UserDefinedType

UserDefinedType se utiliza para establecer tipos de datos (por ejemplo, una cadena) y restricciones XSD. Por ejemplo, con un ejemplar de un UserDefinedType, se puede definir la longitud mínima de una cadena.

La figura 7 visualiza el modelo del esquema del marco de interacción, incluyendo todas sus referencias.

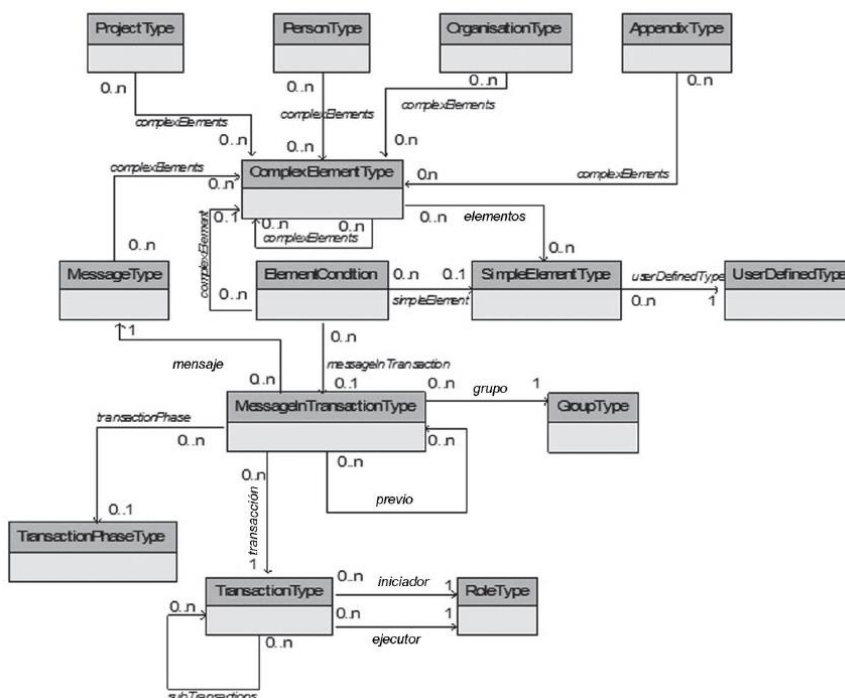


Figura 7 – Tipos y referencias del esquema del marco de interacción

Anexo A (Normativo)

Definición del esquema del marco de interacción

A.1 Introducción

Un marco de interacción ha de cumplir con las reglas que se incluyen en un esquema de marco de interacción. En este anexo, la definición del esquema de la estructura de interacción se da en formato EXPRESS y en formato XSD. Además, se describen tipos de elementos, atributos, elementos y referencias.

Todos los objetos en un marco de interacción requieren una fecha de inicio y una fecha de finalización. Este requisito permite limitar el tiempo de validez de un determinado objeto.

A.2 Definición del esquema de marco de interacción (formato EXPRESS)

SCHEMA ISO 229481_Part_2A;

ENTITY ProjectType; - La definición de un grupo específico de proyectos. Generalmente solo estará presente un ejemplar en un marco de interacción que defina la estructura la estructura de los elementos que debería exponer cada ejemplar de proyecto.

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; - La definición de un grupo específico de personas. Generalmente solo estará presente un ejemplar en un marco de interacción que defina la estructura de los elementos que debería exponer cada ejemplar de organización.

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; - La definición de un grupo específico de organizaciones. Generalmente solo estará presente un ejemplar en un marco de interacción que defina la estructura de los elementos que debería exponer cada ejemplar de persona.

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; - Un AppendixType contiene la definición de un archivo adjunto. Los elementos de datos que se debería registrar con un archivo se pueden especificar en la sección de elementos complejos.

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; - Un ComplexElementType contiene un juego de SimpleElementTypes. Cada SimpleElementType declarado se da exactamente el número de veces que se menciona.

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; - La definición de un tipo de mensaje (MessageType), especifica la estructura del mensaje y qué juego de SimpleElementType (a través de ComplexElementType) puede acompañar.

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; - La condición de cómo se utiliza un SimpleElementType dentro de un tipo de mensaje.

description: STRING;

condition: STRING;

helpInfo: OPTIONAL STRING;

complexType: OPTIONAL ComplexElementType;

simpleElement: OPTIONAL SimpleElementType;

messageInTransaction: OPTIONAL MessageInTransactionType;

END_ENTITY;

ENTITY SimpleElementType; - La definición de un tipo de elemento simple (SimpleElementType). Este tipo de elemento especifica una propiedad que se puede dar dentro de varias estructuras de objetos, por ejemplo, en MessageType (véase también AppendixType, ProjectType, PersonType y OrganizationType). Un SimpleElementType siempre está incrustado en un 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; - Una especificación de un tipo de datos (que se utilizará en un SimpleElementType). Este tipo de datos encapsula áreas de relleno en el mensaje final, por ejemplo, un código postal holandés comienza siempre con cuatro dígitos y luego dos caracteres.

description: STRING;

state: STRING;

dateLaMu: DATETIME;

userLaMu: STRING;

baseType: STRING;

xsdRestriction: OPTIONAL STRING;

language: OPTIONAL STRING;

helpInfo: OPTIONAL STRING;

END_ENTITY;

ENTITY MessageInTransactionType; - La participación de un MessageType dentro de un TransactionType relacionado con un cierto tipo de grupo (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; - La definición de una fase relacionada con una transacción. Algunos ejemplos pueden ser “asignación aceptada” o “parte de resultado

recibida”.

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; - La definición de un tipo de transacción. Un tipo de transacción puede hacer referencia a otros tipos de transacción. Una transacción será iniciada por una persona perteneciente a una organización que desempeñe un determinado rol. En este nivel, el tipo de rol del iniciador debería declararse (que lo hará cumplir el esquema promocionado). Lo mismo ocurre con el ejecutor.

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; - La definición de un tipo de rol específico.

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; - La definición de un grupo para almacenar archivos adjuntos enviados con un mensaje dentro de una transacción.

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 Definición del esquema de marco de interacción (formato XSD)

```
<?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: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 alway 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:element>
          </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" >
            <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:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="MessageInTransactionTypeType" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementType" >
            <xsd:sequence>
                <xsd:element name="requiredNotify" type = "xsd:integer"/
>
                <xsd:element name="dateLaMu" type = "xsd:dateTime"/ >
                <xsd:element name="userLaMu" type = "xsd:string"/ >
                <xsd:element name="received" type = "xsd:boolean"/ >
                <xsd:element name="send" type = "xsd:boolean"/ >
                <xsd:element name="state" type = "xsd:string"/ >
                <xsd:element name="initiatorToExecutor" type =
"xsd:boolean" minOccurs = "0"/ >
                <xsd:element name="openSecondaryTransactionsAllowed" type
= "xsd:boolean" minOccurs = "0"/ >

```

```

minOccurs = "0"/ >
    <xsd:element name="firstMessage" type = "xsd:boolean"
    <xsd:element name="message" >
        <xsd:complexType>
            <xsd:choice>
                <xsd:element ref="iso29481p2a:MessageType"/ >
                <xsd:element
ref="iso29481p2a:MessageTypeRef"/ >
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="previous" minOccurs = "0" >
        <xsd:complexType>
            <xsd:choice maxOccurs="unbounded" >
                <xsd:element
ref="iso29481p2a:MessageInTransactionType"/ >
                <xsd:element
ref="iso29481p2a:MessageInTransactionTypeRef"/ >
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="transaction" >
        <xsd:complexType>
            <xsd:choice>
                <xsd:element
ref="iso29481p2a:TransactionType"/ >
                <xsd:element
ref="iso29481p2a:TransactionTypeRef"/ >
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="transactionPhase" minOccurs = "0" >
        <xsd:complexType>
            <xsd:choice>
                <xsd:element
ref="iso29481p2a:TransactionPhaseType"/ >
                <xsd:element
ref="iso29481p2a:TransactionPhaseTypeRef"/ >
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="group" >
        <xsd:complexType>
            <xsd:choice>
                <xsd:element ref="iso29481p2a:GroupType"/ >
                <xsd:element ref="iso29481p2a:GroupTypeRef"/
>
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="appendixTypes" minOccurs = "0" >
        <xsd:complexType>
            <xsd:choice maxOccurs="unbounded" >
                <xsd:element ref="iso29481p2a:AppendixType"/
>
            <xsd:element
ref="iso29481p2a:AppendixTypeRef"/ >

```

```

        </xsd:choice>
      </xsd:complexType>
    </xsd:element>
  </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="MessageTypeType" >
  <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:element>
        <xsd:element name="appendixTypes" minOccurs = "0" >
          <xsd:complexType>
            <xsd:choice maxOccurs="unbounded" >
              <xsd:element ref="iso29481p2a:AppendixType"/
>
            </xsd:choice>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="OrganisationTypeType" >
  <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:element>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonTypeType" >
    <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:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ProjectTypeType" >

```

```

    <xsd:complexContent>
      <xsd:extension base="iso29481p2a:elementType" >
        <xsd:sequence>
          <xsd:element name="namespace" type = "xsd:string"/ >
          <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:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="RoleTypeType" >
    <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="responsibilityScope" type =
"xsd:string" minOccurs = "0"/ >
          <xsd:element name="responsibilityTask" type =
"xsd:string" minOccurs = "0"/ >
          <xsd:element name="responsibilitySupportTask" type =
"xsd:string" minOccurs = "0"/ >

```

```

        <xsd:element name="responsibilityFeedback" type =
"xsd:string" minOccurs = "0"/ >
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SimpleElementType" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementType" >
            <xsd:sequence>
                <xsd:element name="description" type = "xsd:string"/ >
                <xsd:element name="interfaceType" type = "xsd:string"/ >
                <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="valueList" type = "xsd:string"
minOccurs = "0"/ >
                <xsd:element name="userDefinedType" >
                    <xsd:complexType>
                        <xsd:choice>
                            <xsd:element
ref="iso29481p2a:UserDefinedType"/ >
                            <xsd:element
ref="iso29481p2a:UserDefinedTypeRef"/ >
                        </xsd:choice>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TransactionPhaseType" >
    <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:sequence>
        </xsd:extension>
    </xsd:complexContent>

```

```

    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="TransactionTypeType" >
    <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="result" type = "xsd:string" minOccurs =
= "0"/ >
          <xsd:element name="subTransactions" minOccurs = "0" >
            <xsd:complexType>
              <xsd:choice maxOccurs="unbounded" >
                <xsd:element
ref="iso29481p2a:TransactionType"/ >
                <xsd:element
ref="iso29481p2a:TransactionTypeRef"/ >
              </xsd:choice>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="initiator" >
            <xsd:complexType>
              <xsd:choice>
                <xsd:element ref="iso29481p2a:RoleType"/ >
                <xsd:element ref="iso29481p2a:RoleTypeRef"/ >
              </xsd:choice>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="executor" >
            <xsd:complexType>
              <xsd:choice>
                <xsd:element ref="iso29481p2a:RoleType"/ >
                <xsd:element ref="iso29481p2a:RoleTypeRef"/ >
              </xsd:choice>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="appendixTypes" minOccurs = "0" >
            <xsd:complexType>
              <xsd:choice maxOccurs="unbounded" >
                <xsd:element ref="iso29481p2a:AppendixType"/
>
                <xsd:element
ref="iso29481p2a:AppendixTypeRef"/ >
              </xsd:choice>
            </xsd:complexType>

```

```

        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="UserDefinedTypeType" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementType" >
      <xsd:sequence>
        <xsd:element name="description" type = "xsd:string"/ >
        <xsd:element name="state" type = "xsd:string"/ >
        <xsd:element name="dateLaMu" type = "xsd:dateTime"/ >
        <xsd:element name="userLaMu" type = "xsd:string"/ >
        <xsd:element name="baseType" type = "xsd:string"/ >
        <xsd:element name="xsdRestriction" type = "xsd:string"
minOccurs = "0"/ >
        <xsd:element name="language" type = "xsd:string"
minOccurs = "0"/ >
        <xsd:element name="helpInfo" type = "xsd:string"
minOccurs = "0"/ >
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- complex types for element reference declarations (for ENTITY
definitions) ->
<xsd:complexType name="AppendixTypeTypeRef" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementTypeRef"/ >
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ComplexElementTypeTypeRef" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementTypeRef"/ >
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ElementConditionTypeRef" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementTypeRef"/ >
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="GroupTypeTypeRef" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementTypeRef"/ >
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="MessageInTransactionTypeTypeRef" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementTypeRef"/ >
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="MessageTypeTypeRef" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementTypeRef"/ >
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="OrganisationTypeTypeRef" >

```

```

        <xsd:complexContent>
            <xsd:extension base="iso29481p2a:elementTypeRef"/ >
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="PersonTypeTypeRef" >
        <xsd:complexContent>
            <xsd:extension base="iso29481p2a:elementTypeRef"/ >
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="ProjectTypeTypeRef" >
        <xsd:complexContent>
            <xsd:extension base="iso29481p2a:elementTypeRef"/ >
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="RoleTypeTypeRef" >
        <xsd:complexContent>
            <xsd:extension base="iso29481p2a:elementTypeRef"/ >
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="SimpleElementTypeTypeRef" >
        <xsd:complexContent>
            <xsd:extension base="iso29481p2a:elementTypeRef"/ >
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="TransactionPhaseTypeTypeRef" >
        <xsd:complexContent>
            <xsd:extension base="iso29481p2a:elementTypeRef"/ >
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="TransactionTypeTypeRef" >
        <xsd:complexContent>
            <xsd:extension base="iso29481p2a:elementTypeRef"/ >
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="UserDefinedTypeTypeRef" >
        <xsd:complexContent>
            <xsd:extension base="iso29481p2a:elementTypeRef"/ >
        </xsd:complexContent>
    </xsd:complexType>
    <!-- group declarations (for SELECT data type definitions) -->
    <!-- enumeration type declarations (for ENUMERATION data type definitions)
→
    <!-- simple type declarations (for TYPE defined data type definitions) -->
    <!-- standard declarations, elementType, simpleType, logical (for each
translation) -->
    <xsd:complexType name="elementType" >
        <xsd:attribute name="id" type = "xsd:ID" use = "required"/ >
    </xsd:complexType>
    <xsd:complexType name="elementTypeRef" >
        <xsd:attribute name="idref" type = "xsd:IDREF" use = "required"/ >
    </xsd:complexType>
    <xsd:simpleType name="logical" >
        <xsd:restriction base="xsd:string" >
            <xsd:enumeration value="true"/ >
            <xsd:enumeration value="false"/ >
            <xsd:enumeration value="unknown"/ >
        </xsd:restriction>

```

```
</xsd:simpleType>  
</xsd:schema>
```

A.4 Tipos de elementos

A.4.1 AppendixType

Atributos: id

Elementos: description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpInfo, code

Referencias: complexElements

Un AppendixType contiene la definición de un archivo adjunto. Los elementos de datos que se debería registrar con un archivo adjunto se pueden especificar en la sección de elementos complejos.

EJEMPLO

```
<AppendixType id="StandardAppendixType" >  
  <description>Standard appendix type</description>  
  <startDate>2007-01-23T00:00:00Z</startDate>  
  <endDate>2007-12-31T00:00:00Z</endDate>  
  <state>active</state>  
  <dateLaMu>2007-01-23T00:00:00Z</dateLaMu>  
  <userLaMu>bapa</userLaMu>  
</AppendixType>
```

Nota: Referencia appendixType. Si un marco de interacción especifica muchos tipos de archivos adjuntos, puede ser difícil para los usuarios seleccionar el tipo de archivo adjunto apropiado en determinadas situaciones. Esta referencia filtra el conjunto de todos los tipos de archivos adjuntos a los que son válidos para esta transacción o mensaje.

A.4.2 ComplexElementType

Atributos: id

Elementos: description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpInfo

Referencias: elements

Un tipo de elemento complejo contiene un conjunto de SimpleElementTypes. Cada uno de los elementos declarados de ElementType se da exactamente el número de veces que se menciona.

EJEMPLO

```
<ComplexElementType id="MenuItem" >
```

```

<description>Item on menu</description>
<startDate>2007-01-23T00:00:00Z</startDate>
<endDate>2007-12-31T00:00:00Z</endDate>
<state>active</state>
<dateLaMu>2007-01-23T00:00:00Z</dateLaMu>
<userLaMu>bapa</userLaMu>
<elements>
  <SimpleElementTypeRef idref="Name"/ >
  <SimpleElementTypeRef idref="Price"/ >
  <SimpleElementTypeRef idref="Description"/ >
  <SimpleElementTypeRef idref="Calories"/ >
</elements>
</ComplexElementType>

```

A.4.3 ElementCondition

Atributos: id

Elementos: description, requiredNotify, minValue, maxValue, format, helpInfo

Referencias: message, element

La condición de un SimpleElementType se usa dentro de un MessageType específico.

EJEMPLO

```

<ElementCondition id="Price restriction" >
  <description>Minimal price of a menu item</description>
  <requiredNotify>0</requiredNotify>
  <minValue>5.00</minValue>
  <element>
    <SimpleElementType>Price</SimpleElementType>
  </element>
  <message>
    <MessageTypeRef idref="ProvideWithMenuMessage"/ >
  </message>
</ElementCondition>

```

A.4.4 GroupType

Atributos: id

Elementos: description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpInfo

La definición de un grupo para enviar archivos adjuntos enviados en un mensaje en el seno de una transacción.

EJEMPLO

```

<GroupType id="StandardGroupType" >
  <description>Standard group</description>
  <startDate>2007-12-20T00:00:00Z</startDate>

```

```

<endDate>2008-12-31T00:00:00Z</endDate>
<state>active</state>
<dateLaMu>2007-12-20T00:00:00Z</dateLaMu>
<userLaMu>bapa</userLaMu>
</GroupType>

```

A.4.5 MessageType

Atributos: id

Referencias: complexElements, appendixTypes

La definición de un tipo de mensaje (MessageType), la cual especifica la estructura del mensaje y qué conjunto de SimpleElementType (a través de ComplexElementType) puede incorporar.

EJEMPLO

```

<MessageType id="ProvideWithMenuMessage" >
  <description>Message that contains the menu.</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <complexElements>
    <ComplexElementTypeRef idref="Menu"/ >
  </complexElements>
  <appendixTypes>
    <AppendixTypeRef idref="Menucard" / >
  </appendixTypes>
</MessageType>

```

Nota: Elemento **firstmessage**. En general, una transacción será iniciada por el rol iniciador satisfaciendo a la persona con un tipo de mensaje que es independiente de cualquier tipo de mensaje anterior. Sin embargo, este principio no puede aplicarse a una transacción que actúa como sub-transacción de otra transacción. Este elemento indica explícitamente si se puede utilizar un tipo de mensaje para iniciar una nueva transacción.

A.4.6 MessageInTransactionType

Atributos: id

Elementos: message, previous, transaction, transactionPhase, group, appendixTypes

La participación de un MessageType dentro de un TransactionType relacionado con un cierto tipo de grupo (GroupType).

EJEMPLO

```

<MessageInTransactionType id="MiTT_002" >
  <requiredNotify>0</requiredNotify>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>

```

```

<userLaMu>bapa</userLaMu>
<received>true</received>
<send>true</send>
<state>active</state>
<initiatorToExecutor>false</initiatorToExecutor>
<openSecondaryTransactionsAllowed>true</openSecondaryTransactionsAllowed>
<firstMessage>false</firstMessage>
<message>
  <MessageTypeRef idref="ProvideWithMenuMessage"/ >
</message>
<previous>
  <MessageInTransactionTypeRef idref="MiTT_001"/ >
</previous>
<transaction>
  <TransactionTypeRef idref="ObtainMenuTransaction"/ >
</transaction>
<transactionPhase>
  <TransactionPhaseTypeRef idref="MenuDelivered" >
</transactionPhase>
<group>
  <GroupTypeRef idref="StandardGroupType"/ >
</group>
<appendixTypes>
  <AppendixTypeRef idref="Menucard" / >
</appendixTypes>
</MessageInTransactionType>

```

A.4.7 OrganisationType

Atributos: id

Elementos: description, startDate, endDate, state, dateLaMu, language, category, helpInfo, code

Referencia: complexElements

La definición de un grupo específico de organizaciones. Generalmente, sólo un ejemplar estará presente en un marco de interacción que defina la estructura de los elementos que debería exponer cada ejemplar de organización.

EJEMPLO

```

<OrganisationType id="StandardOrganisationType" >
  <description>Standard organisation type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</OrganisationType>

```

A.4.8 PersonType

Atributos: id

Elementos: description, startDate, endDate, state, dateLaMu, language, category, helpInfo, code

Referencias: complexElements

La definición de un grupo específico de organizaciones. Generalmente, sólo un ejemplar estará presente en un marco de interacción que defina la estructura de los elementos que debería exponer cada ejemplar de organización.

EJEMPLO

```
<PersonType id="StandardPersonType" >
  <description>Standard person type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</PersonType>
```

A.4.9 ProjectType

Atributos: id

Elementos: namespace, description, startDate, state, dateLaMu, language, category, helpInfo, code

Referencias: complexElements

La definición de un grupo específico de proyectos. En general, sólo un ejemplar estará presente en un marco de interacción que defina la estructura de los elementos que debería exponer cada ejemplar de proyecto.

EJEMPLO

```
<ProjectType id="StandardProjectType" >
  <namespace>http://www.ISO_29481_Part_2A.com/testproject</namespace>
  <description>Standard project type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</ProjectType>
```

A.4.10 RoleType

Atributos: id

Elementos: namespace, description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpInfo, code

Referencias: complexElements

La definición de un grupo específico de proyectos. En general, sólo un ejemplar estará presente en un marco de interacción que defina la estructura de los elementos que debería exponer cada ejemplar de proyecto.

EJEMPLO

```
<RoleType id="waiter" >
  <description>Responsible for taking and posting of orders</description>
  <startDate>2008-05-04T00:00:00:00Z</startDate>
  <endDate>2008-05-04T00:00:00.00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-05-04T00:00:00.00Z</dateLaMu>
  <userLaMu>MMA</userLaMu>
  <language/>
  <category/>
  <helpInfo/>
  <code/>
  <responsibilityScope/>
  <responsibilityTask/>
  <responsibilitySupportTask/>
  <responsibilityFeedback/>
</RoleType>
```

A.4.11 SimpleElementType

Atributes: id

Elements: description, interfaceType, state, dateLaMu, userLaMu, language, category, helpInfo, valueList

References: composedOf, userDefinedType

La definición de un tipo de elemento simple (SimpleElementType). Este tipo de elemento especifica una propiedad que puede darse dentro de varias estructuras de objetos, por ejemplo, en MessageType (véase también AppendixType, ProjectType, PersonType y OrganizationType). Un SimpleElementType siempre está incrustado en un ComplexElementType.

EJEMPLO

```
<SimpleElementType id="Name" >
  <description>Name of the menu item</description>
  <interfaceType/>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <userDefinedType>
    <UserDefinedTypeRef idref="String"/ >
  </userDefinedType>
</SimpleElementType>
```

A.4.12 TransactionPhaseType

Atributos: id

Elementos: description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpInfo, code

Referencias: ComposedOf, userDefinedType

La definición de una fase relacionada con una transacción. Algunos ejemplos pueden ser "asignación aceptada" o "parte de resultado recibido".

EJEMPLO

```
<TransactionPhaseType id="WaitingForMenu" >
  <description>Menu requested but not yet delivered</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</TransactionPhaseType>
```

A.4.13 TransactionType

Atributos: id

Elementos: description, startDate, endDate, state, dateLaMu, language, category, helpInfo, code, result

Referencias: interior, executor, subTransactions

La definición de un tipo de transacción. Un tipo de transacción puede hacer referencia de nuevo a otros tipos de transacción. Una transacción será iniciada por una persona perteneciente a una organización que desempeñe un determinado papel. En este punto se debería declarar el tipo de rol del iniciador (el esquema refuerza este concepto). Lo mismo ocurre en el ejecutor.

EJEMPLO

```

<TransactionType id="MenuObtainingTransaction" >
  <description>The transaction to obtain the proper menu</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <initiator>
    <RoleTypeRef idref="Customer"/ >
  </initiator>
  <executor>
    <RoleTypeRef idref="Employee"/ >
  </executor>
</TransactionType>

```

A.4.14 UserDefinedType**Attributes:** #id**Elements:** description, state, dateLaMu, baseType, xsdRestriction, language, helpInfo

Una especificación de un tipo de datos (que se utilizará en un SimpleElementType). Este tipo de datos encapsula áreas de relleno en el mensaje final.

EJEMPLO

```

<UserDefinedType id="String" >
  <description>Standard string</description>
  <state>active</state>
  <dateLaMu>2007-12-20T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <baseType>STRING</baseType>
  <xsdRestriction/>
</UserDefinedType>

```

A.5 Attributes**A.5.1 id**

Nombre "corto" único evento. Después de la promoción, este nombre será un nombre de objeto.

EJEMPLO

```

Framework object
<OrganisationType id="TNO Building and Construction" >
  <description>The attributes of an organisation</description>
  <startDate>2007-12-12T00:00:00Z</startDate>
  <endDate>9999-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2007-12-12T00:00:00Z</dateLaMu>

```

```

    <userLaMu>testmanagement</userLaMu>
    <language>NL</language>
    <category>S</category>
    <helpInfo>http://.../</helpInfo>
    <code>DEFAULT</code>
</OrganisationType>
Message object
    <name>TNO Building & Construction</name>
    <state>active</state>
    <dateLaMu>2007-12-02T00:00:00Z</dateLaMu>
    <userLaMu>Peter Bonsma</userLaMu>
</Organisation>

```

A.6 Elements

A.6.1 baseType

Guarda el tipo de base de un SimpleElementType.

EJEMPLO

```

<SimpleElementType id="Height" >
    ...
    <userDefinedType>
        <UserDefinedType id="..." >
            ...
            <baseType>INTEGER</baseType>
            ...
        </UserDefinedType>
    </userDefinedType>
</SimpleElementType>

```

Aquí el tipo de base de un SimpleElementType.

A.6.2 category

Categoría asociada con este objeto (valor opcional).

A.6.3 code

Código de marco de interacción acordado para este objeto.

EJEMPLO

```

<... id="..." >
    ...
    EAN 33156
    ...
</...>

```

A.6.4 dateLaMu

Día y hora de la última mutación de este objeto.

```
<... id="..." >
  ...
  <dateLaMu>2007-12-02T00:00:00Z</dateLaMu>
  ...
</...>
```

A.6.5 description

Descripción del objeto.

EJEMPLO

```
<... id="Doorleaf" >
  ...
  <description>The leaf of a flat door.</description>
  ...
</...>
```

A.6.6 endDate

Día y hora final de validación de este objeto.

EJEMPLO

```
<... id="..." >
  ...
  <endDate>2007-02-03T00:00:00Z</endDate>
  ...
</...>
```

A.6.7 firstMessage

Valor booleano opcional para indicar que este objeto MessageInTransactionType se puede utilizar como un mensaje de inicio para una nueva transacción. El valor predeterminado es falso.

EJEMPLO

```
<MessageInTransactionType id="..." >
  ...
  <firstMessage>true</firstMessage>
  ...
</MessageInTransactionType>
```

A.6.8 format

La disposición de un elemento (opcional).

A.6.9 helpInfo

Vínculo URL/URI para mayor información sobre este objeto.

EJEMPLO

```
<... id="..." >
  ...
  <helpInfo> http://www.ISO
29481_Part_2A.com/helpInfo_object0001.html</helpInfo>
  ...
</...>
```

A.6.10 initiatorToExecutor

Valor booleano que representa la dirección a la que se supone que se va a enviar el mensaje.

EJEMPLO

```
<MessageInTransactionType id="..." >
  ...
  <initiatorToExecutor>false</initiatorToExecutor>
  ...
  <message>
    <MessageType id="TenderAcceptance" >
      ...
    </MessageType>
  </message>
  <transaction>
    <TransactionType id="TenderTraject" >
      ...
      <initiator>
        <RoleType id="Performer" >
          ...
        </RoleType>
      </initiator>
      <executor>
        <RoleType id="Client" >
          ...
        </RoleType>
      </executor>
    </TransactionType>
  </transaction>
  ...
</MessageInTransactionType id="..." >
```

Se prevé que el mensaje TenderAcceptance se enviará desde el cliente (ejecutor) al actor (iniciador).

A.6.11 interfaceType

Tipo interfaz o vista en este SimpleElementType para un mensaje específico.

A.6.12 language

Lenguaje a usar tras la promoción de este objeto.

EJEMPLO

```
<... id="..." >
  ...
  <language>NL</language>
  ...
</...>
```

A.6.13 namespace

Nombre de destino del espacio de nombre para identificar los mensajes que pertenecen a este marco de interacción.

EJEMPLO

```
<ProjectType id="..." >
  <namespace>http://www.visi.nl/testraamwerk</namespace>
  ...
</ProjectType>
```

Nota: Espacio de nombres de elemento. Anteriormente, todos los esquemas de la estructura de interacción se referían al mismo espacio de nombres de destino, que pasó a ser el mismo que el espacio de nombres del propio esquema de interacción. Utilizando este elemento, cada esquema de interacción puede especificar su propio espacio de nombres.

A.6.14 openSecondaryTransactionsAllowed

Valor booleano opcional que permite continuación de la transacción primaria incluso si no se completan todas las transacciones secundarias. Un valor "TRUE" se interpreta como una luz verde para la continuación (puerta OR). Un valor "FALSE" se interpreta como una luz roja para continuar como una luz roja para continuar hasta que se completan todas las transacciones secundarias (puerta AND). Si no se especifica openSecondaryTransactionsAllowed, la interpretación equivale a "TRUE".

A.6.15 received

Valor booleano que indica que el mensaje anterior se ha recibido.

A.6.16 requiredNotify

No se ha asociado ningún significado a este elemento.

A.6.17 responsibilityFeedback

Respuesta esperada en la dirección de otros roles (cadena opcional).

A.6.18 responsibilityScope

Alcance/marco de las responsabilidades definidas del rol (cadena opcional).

A.6.19 responsibilitySupportTask

Tareas para ser ejecutadas con el objeto de apoyar otros roles. Por ejemplo, la delegación de responsabilidades (cadena opcional).

A.6.20 responsibilityTask

Tareas originadas por las responsabilidades de este rol (cadena opcional).

A.6.21 result

Resultado esperado por la ejecución de la transacción.

A.6.22 send

Valor booleano que indica si el mensaje ha sido enviado o no.

A.6.23 startDate

Fecha y hora de inicio de la validación de este objeto.

EJEMPLO

```
<... id="..." >
  ...
  <startDate>2007-02-03T00:00:00Z</startDate>
  ...
</...>
```

A.6.24 state

Estado de visibilidad de este objeto, valores posibles:

EJEMPLO

```
<... id="..." >
  ...
  <state>active</state>
  ...
</...>
```

y

```
<... id="..." >
  ...
  <state>passive</state>
  ...
</...>
```

A.6.25 userLaMu

Usuario de la última mutación de este objeto.

EJEMPLO

```
<... id="..." >
  ...
  <state>passive</state>
  ...
</...>
```

A.6.26 valueList

Lista de valores separados por punto y coma que puede tomar un evento a nivel de mensaje. Originalmente, este elemento se destinaba a apoyar enumeraciones. Actualmente, esto se implementa con el tipo de elemento UserDefinedType y el elemento xsdRestriction. En xsdRestriction, se especifican los valores de enumeración. No se vincula ningún significado a este elemento.

EJEMPLO

```
<... id="..." >
  ...
  <state>passive</state>
  ...
</...>
```

A.6.27 xsdRestriction

Este elemento especifica una restricción que se realizará a nivel de mensaje en el SimpleElementType de este UserDefinedType.

A.7 References**A.7.1 appendixTypes**

Conjunto de tipos de archivos adjuntos seleccionables.

A.7.2 complexElements

Referencia a un conjunto de SimpleElementType (recogido en un ComplexElementType).

EJEMPLO

```
<... id="Abc" >
  ...
  <complexElements>
    <ComplexElementType id="ElementsSet1" >
      ...
    </ComplexElementType>
  </complexElements>
</...>
```

```

    <elements>
      <SimpleElementType id="Element_A" >
        ...
      </SimpleElementType>
      <SimpleElementType id="Element_B" >
        ...
      </SimpleElementType>
    </elements>
  </ComplexElementType>
  <ComplexElementType id="ElementsSet2" >
    ...
  </ComplexElementType>
  <ComplexElementType id="ElementsSet3" >
    ...
  </ComplexElementType>
</complexElements>
</...>

```

A nivel de mensaje, esto se elabora en:

```

<Abc id="..." >
  ...
  <elementsSet1>
    <ElementsSet1>
      <Element_A>...</Element_A>
      <Element_B>...</Element_B>
    </ElementsSet1>
    ...
    <ElementsSet1>
      <Element_A>...</Element_A>
      <Element_B>...</Element_B>
    </ElementsSet1>
  </elementsSet1>
  <elementsSet2>
    <ElementsSet2>
      ...
    </ElementsSet2>
    ...
    <ElementsSet2>
      ...
    </ElementsSet2>
  </elementsSet2>
  <elementsSet3>
    <ElementsSet3>
      ...
    </ElementsSet3>
    ...
    <ElementsSet3>
      ...
    </ElementsSet3>
  </elementsSet3>
</...>

```

A.7.3 composedOf

Un SimpleElementType puede consistir en un conjunto de SimpleElementType (recogido en un (ComplexElementType)).

EJEMPLO

```
<SimpleElementType id="Doorleaf" >
  ...
  <composedOf>
    <ComplexElementType id="Measurement" >
      ...
      <elements>
        <SimpleElementType id="Height" >
          ...
        </SimpleElementType>
        <SimpleElementType id="Width" >
          ...
        </SimpleElementType>
        <SimpleElementType id="Thickness" >
          ...
        </SimpleElementType>
      </elements>
    </ComplexElementType>
    <ComplexElementType id="Material-selection" >
      ...
      <elements>
        <SimpleElementType id="Type-of-wood" >
          ...
        </SimpleElementType>
      </elements>
    </ComplexElementType>
  </composedOf>
</SimpleElementType>
```

A nivel de mensaje, esto se elabora en:

```
<Doorleaf>
  ...
  <Measurement id="..." >
    <Height>...</Height>
    <Width>...</Width>
    <Thickness>...</Thickness>
  </Measurement>
  <Material-selection id="..." >
    <Type-of-wood>...</Type-of-wood>
  </Material-selection>
</Doorleaf>
```

A.7.4 element

La condición en un SimpleElementType que se utilizará en un MessageType

EJEMPLO

```
<ElementCondition id="..." >
  ...
  <minValue>2000</minValue>
  ...
  <element>
    <SimpleElementTypeRef idref="Doorheight" >
  </element>
  <message>
    <MessageType id="M" >
      ...
      <complexElements>
        <ComplexElementType>
          ...
          <elements>
            <SimpleElementType id="Doorheight" >
              ...
            </SimpleElementType>
          </elements>
        </ComplexElementType>
      </complexElements>
    </MessageType>
  </message>
</ElementCondition>
```

Se especifica que dentro de MessageType, el valor M, es la altura de la puerta (*Doorheight*) debería ser al menos 2000, aunque esto no se aplica a nivel SimpleElementType.

A.7.5 elements

Conjunto de SimpleElementType disponible dentro de un ComplexElementType.

EJEMPLO

```
<ComplexElementType id="Door" >
  ...
  <elements>
    <SimpleElementType id="Doorleaf" >
      ...
    </SimpleElementType>
    <SimpleElementType id="Fastenings" >
      ...
    </SimpleElementType>
    <SimpleElementType id="UpperWindow" >
      ...
    </SimpleElementType>
  </elements>
</ComplexElementType>
```

A nivel de mensaje, esto se elabora en:

```
<... id="..." >
...
<door>
  <Door>
    <Doorleaf>...</Doorleaf>
    <Fastenings>...</Fastenings>
    <UpperWindow>...</UpperWindow>
  </Door>
  ...
  <Door>
    <Doorleaf>...</Doorleaf>
    <Fastenings>...</Fastenings>
    <UpperWindow>...</UpperWindow>
  </Door>
</door>
</...>
```

Es obligatorio que todos los elementos se expresen con precisión sólo una vez. Estas puertas siempre tienen una ventana superior. El número de puertas no tiene restricciones.

A.7.6 executor

Es el rol de ejecutor de una tarea (RoleType) asociado con una transacción particular.

EJEMPLO

```
<TransactionType id="TenderTraject" >
...
<executor>
  <RoleType id="Client" >
    ...
  </RoleType>
</executor>
...
</TransactionType>
```

A.7.7 group

El GroupType que está asociado con un mensaje dentro de una transacción particular.

EJEMPLO

```
<MessageInTransactionType id="..." >
...
<message>
  <MessageType id="M" >
    ...
  </MessageType>
</message>
<group>
  <GroupType id="G" >
```

```

    ...
    </GroupType>
  </group>
  <transaction>
    <TransactionType id="T" >
      ...
      </TransactionType>
    </transaction>
    ...
  </MessageInTransactionType>

```

El mensaje M dentro de la transacción T pertenece al grupo G (puede haber más mensajes M dentro de la transacción T que pertenezcan al mismo o a otro grupo).

A.7.8 iniciador

Es el rol de iniciador de una tarea (RoleType) asociado con una transacción particular.

EJEMPLO

```

<TransactionType id="TenderTraject" >
  ...
  <initiator>
    <RoleType id="Performer" >
      ...
    </RoleType>
  </initiator>
  ...
</TransactionType>

```

A.7.9 message

Es el mensaje que hay en un MessageInTransactionType.

EJEMPLO

```

<MessageInTransactionType id="..." >
  ...
  <message>
    <MessageType id="..." >
      ...
    </MessageType>
  </message>
  ...
</MessageInTransactionType>

```

A.7.10 previous

Un MessageInTransactionType puede reforzar que un mensaje previo dentro de una transacción particular debería ejecutarse (este MessageInTransactionType anterior no necesita pertenecer al mismo TransactionType).

EJEMPLO

```

<MessageInTransactionType id="..." >
  ...
  <previous>
    <MessageInTransactionType id="..." >
      ...
      <message>
        <MessageType id="Tender" >
          ...
          </MessageType>
        </message>
      ...
    </MessageInTransactionType>
  </previous>
  ...
  <message>
    <MessageType id="TenderAcceptance" >
      ...
      </MessageType>
    </message>
    ...
  </MessageInTransactionType>subTransactions

```

Transacciones que podrían operar en esta transacción.

EJEMPLO

```

<TransactionType id="AcquireWork" >
  ...
  <subTransactions>
    <TransactionType id="AcquisitionTrack" >
      ...
      </TransactionType>
    <TransactionType id="TenderTrack" >
      ...
      </TransactionType>
    </subTransactions>
  </TransactionType>

```

El TransactionType *AcquireWork* consta de la *AcquisitionTrack* y de la *TenderTraject* de la TransactionType.

A.7.11 simpleElements

Conjunto de tipos de elementos simples pertenecientes a este tipo de elementos complejos.

EJEMPLO

```

<ComplexElementType id="Door" >
  ...
  <simpleElements>
    <SimpleElementType id="Doorleaf" >
      ...
    </SimpleElementType>
  </simpleElements>

```

```

    <SimpleElementType id="HingesAndLocks" >
    ...
  </SimpleElementType>
  <SimpleElementType id="UpperWindow" >
    ...
  </SimpleElementType>
</simpleElements>
</ComplexElementType>

```

A nivel de mensaje, esto puede conducir a:

```

<... id="..." >
  ...
  <door>
    <Door>
      <Doorleaf>...</Doorleaf>
      < HingesAndLocks >...</ HingesAndLocks >
      < UpperWindow >...</ UpperWindow >
    </Door>
    ...
    <Door>
      < Doorleaf >...</ Doorleaf >
      < HingesAndLocks >...</ HingesAndLocks >
      < UpperWindow >...</ UpperWindow >
    </Door>
  </door>
</...>

```

A.7.12 subTransactions

Transacciones que se pueden iniciar desde esta transacción.

EJEMPLO

```

<TransactionType id="AcquisitionOfWork" >
  ...
  <subTransactions>
    <TransactionType id="AcquisitionTraject" >
      ...
    </TransactionType>
    <TransactionType id="TenderTraject" >
      ...
    </TransactionType>
  </subTransactions>
</TransactionType>

```

A.7.13 Transaction

Una transacción en un MessageInTransactionType.

EJEMPLO

```
<MessageInTransactionType id="..." >
  ...
  <transaction>
    <TransactionType id="..." >
      ...
    </TransactionType>
  </transaction>
  ...
</MessageInTransactionType>
```

A.7.14 transactionPhase

Una TransactionPhase para un MessageType específico dentro de TransactionType específico.

EJEMPLO

```
<MessageInTransactionType id="..." >
  ...
  <message>
    <MessageType id="M" >
      ...
    </MessageType>
  </message>
  <transaction>
    <TransactionType id="T" >
      ...
    </TransactionType>
  </transaction>
  ...
  <transactionPhase>
    <TransactionPhaseType id="TP" >
      ...
    </TransactionPhaseType>
  </transactionPhase>
  ...
</MessageInTransactionType>
```

Donde un MessageType M dentro de un TransactionType T específico, determina TransactionPhaseType TP.

A.7.15 userDefinedType

Referencia a un UserDefinedType, especifica la forma del SimpleElementType.

EJEMPLO

```
<SimpleElementType id="Height" >
  ...
  <userDefinedType>
    <UserDefinedType id="..." >
      ...
      <baseType>INTEGER</baseType>
      ...
    </UserDefinedType>
  </userDefinedType>
</SimpleElementType>
```

El SimpleElementType *Height* especifica un entero para cada evento (posiblemente con una restricción como xsdRestriction).

Anexo B (Normativo)

Definición de plantillas

B.1 Introducción

Una vez que se dispone de un marco de interacción válido, se necesita un archivo de esquema de interacción que incluya reglas para la comunicación real. Las reglas que son independientes de la comunicación real se incluyen en el archivo de plantillas. En este anexo, se da la definición de plantillas en formato EXPRESS. Además, se describen tipos de elementos, atributos, elementos y referencias.

B.2 Definición de plantillas

SCHEMA ISO 29481_Part_2B;

ENTITY GroupTemplate;

name: STRING;

description: STRING;

creationDate: DATETIME;

startDate: DATETIME;

endDate: DATETIME;

state: STRING;

dateLaMu: DATETIME;

userLaMu: STRING;

versionNo: STRING;

transaction: TransactionTemplate;

END_ENTITY;

ENTITY AppendixGroup;

state: STRING;

dateLaMu: DATETIME;

userLaMu: STRING;

group: GroupTemplate;

END_ENTITY;
ENTITY AppendixTemplate;
name: STRING;
fileLocation: STRING;
fileType: STRING;
fileVersion: STRING;
documentIdentification: STRING;
documentVersion: STRING;
documentReference: STRING;
objectCode: OPTIONAL STRING;
startDate: DATETIME;
endDate: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
language: OPTIONAL STRING;
message: MessageTemplate;
appendixGroup: AppendixGroup;
template: ComplexElementTemplate;
END_ENTITY;
ENTITY MessageTemplate;
identification: STRING;
dateSend: DATETIME;
dateRead: DATETIME;
state: STRING;

dateLaMu: DATETIME;
userLaMu: STRING;
initiatingTransactionMessageID: OPTIONAL STRING;
initiatorToExecutor: BOOLEAN;
messageInTransaction: MessageInTransactionTemplate;
transaction: TransactionTemplate;
template: ComplexElementTemplate;
END_ENTITY;
ENTITY MessageInTransactionTemplate;
identification: STRING;
dateSend: DATETIME;
dateRead: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
END_ENTITY;
ENTITY TransactionTemplate;
number: INTEGER;
name: STRING;
description: STRING;
startDate: DATETIME;
endDate: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;

result: OPTIONAL STRING;
initiator: PersonInRole;
executor: PersonInRole;
project: ProjectTypeInstance;
END_ENTITY;
ENTITY TransactionPhaseTemplate;
name: STRING;
description: STRING;
dateReached: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
transaction: TransactionTemplate;
END_ENTITY;
ENTITY PersonTemplate;
userName: STRING;
name: STRING;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
template: ComplexElementTemplate;
END_ENTITY;
ENTITY OrganisationTemplate;
name: STRING;
abbreviation: STRING;

state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
contactPerson: PersonTemplate;
template: ComplexElementTemplate;
END_ENTITY;
ENTITY PersonInRole;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
successor: OPTIONAL PersonInRole;
substituting: OPTIONAL PersonInRole;
contactPerson: PersonTemplate;
organization: OrganisationTemplate;
role: RoleTemplate;
END_ENTITY;
ENTITY RoleTemplate;
name: STRING;
description: STRING;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
category: OPTIONAL STRING;
END_ENTITY;
ENTITY ProjectTypeInstance;

name: STRING;
description: STRING;
startDate: DATETIME;
endDate: DATETIME;
state: STRING;
dateLaMu: DATETIME;
userLaMu: STRING;
template: ComplexElementTemplate;
END_ENTITY;
ENTITY ComplexElementTemplate;
template: SimpleElementVirtual;
END_ENTITY;
END_SCHEMA;

B.3 Plantillas

B.3.1 AppendixGroup

Atributos: id

Elementos: state, dateLaMu, userLaMu

Referencias: group

La tabla de mapeado para las relaciones n:m entre apéndices y grupos.

EJEMPLO Ejemplo a nivel de mensaje:

```
<AppendixGroup id="AppendixGroup_1" >  
  <state>active</state>  
  <dateLaMu>2008-02-04T00:00:00Z</dateLaMu>  
  <userLaMu>bapa</userLaMu>  
  <group>  
    <StandardGroupType id="..." >  
      ...  
    </StandardGroupType>  
  </group>  
</AppendixGroup>
```

Parte asociada al marco de interacción

```
<GroupType id="StandardGroupType" >
  <description>Standard group</description>
  <startDate>2007-12-20T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2007-12-20T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</GroupType>
```

B.3.2 AppendixTemplate

Atributos: id

Elementos: name, fileLocation, filetype, fileVersion, documentIdentification, documentVersion, documentReference, objectCode, startDate, state, dateLaMu, userLaMu, language

Referencias: appendixGroup, message

Registro de los archivos adjuntos.

EJEMPLO Ejemplpo al nivel de mensaje:

```
<Appendix id="ExampleDocument" >
  <name>Voorbeeld</name>
  <fileLocation>\\srv-bouw\Public\project\docs\msword\</fileLocation>
  <fileType>application/msword</fileType>
  <fileVersion>2007</fileVersion>
  <documentIdentification>345899</documentIdentification>
  <documentVersion>1</documentVersion>
  <documentReference>FG783990</documentReference>
  <startDate>2008-02-04T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-02-04T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <language>EN</language>
  <appendixGroup>
    <AppendixGroup id="..." >
      ...
    </AppendixGroup>
  </appendixGroup>
</Appendix>
```

Parte asociada al marco de interacción

```
<description>Standard appendix definition (no user defined data
fields)</description>
  <startDate>2008-02-04T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-02-04T00:00:00Z</dateLaMu>
```

```

    <userLaMu>bapa</userLaMu>
    <language>NL</language>
</AppendixType>

```

B.3.3 ComplexElementTemplate

Atributos: id

B.3.4 Group Template

Atributos: id

Referencias: transaction

Agrupación de archivos adjuntos de un mensaje para recuperar los documentos asociados.

EJEMPLO Ejemplo al nivel de mensaje:

```

<StandardGroupType id="MenuBackgrounds" >
  <name>Menu Pictures</name>
  <description>A set of background pictures to decorate the
menu</description>
  <creationDate>2008-02-04T00:00:00Z</creationDate>
  <startDate>2008-02-04T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state></state>
  <dateLaMu>2008-02-04T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <versionNo>1</versionNO>
  <transaction>
    <MenuObtainingTransaction id="..." >
      ...
    </MenuObtainingTransaction>
  </transaction>
</StandardGroupType>

```

Parte asociada al marco de interacción

```

<GroupType id="StandardGroupType" >
  <description>Standard group</description>
  <startDate>2007-12-20T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2007-12-20T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</GroupType>
<TransactionType id="MenuObtainingTransaction" >
  <description>The transaction to obtain the proper menu</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>

```

```

<initiator>
  <RoleTypeRef idref="Consumer"/ >
</initiator>
<executor>
  <RoleTypeRef idref="Employee"/ >
</executor>
</TransactionType>

```

B.3.5 MessageInTransactionTemplate

Atributos: id

Elementos: identification, dateSend, dateRead, state, dateLaMu, initiatingTransactionMessageID, initiatorToExecutor

Referencias: transaction, messageInTransaction, template

Un ejemplar de messageType. Esta entidad lleva el intercambio de información real entre Organization Template (organizaciones).

EJEMPLO Ejemplo al nivel de mensaje:

```

<HandOutOfMenuMessage id="a002" >
  <identification>id a002</identification>
  <dateSend>2008-01-23T00:00:00Z</dateSend>
  <dateRead>2008-01-23T00:00:00Z</dateRead>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <initiatingTransactionMessageID>a009</initiatingTransactionMessageID>
  <initiatorToExecutor>false</initiatorToExecutor>
  <messageInTransaction>
    <MessageInTransaction12Ref idref="BiT001"/ >
  </messageInTransaction>
  <transaction>
    <MenuObtainingTransaction id="..." >
      ...
    </MenuObtainingTransaction>
  </transaction>
  <menu>
    <Menu id="..." >
      ...
    </Menu>
    ...
    <Menu id="..." >
      ...
    </Menu>
  </menu>
</HandOutOfMenuMessage>

```

Parte asociada al marco de interacción

```

<TransactionType id="MenuObtainingTransaction" >
  <description>The transaction to obtain the proper menu</description>

```

```

<startDate>2008-01-23T00:00:00Z</startDate>
<endDate>2008-12-31T00:00:00Z</endDate>
<state>active</state>
<dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
<userLaMu>bapa</userLaMu>
<initiator>
  <RoleTypeRef idref="Consumer"/ >
</initiator>
<executor>
  <RoleTypeRef idref="Employee"/ >
</executor>
</TransactionType>
<MessageType id="HandOutOfMenuMessage" >
  <description>Message that contains the menu.</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <complexElements>
    <ComplexElementTypeRef idref="Menu"/ >
  </complexElements>
</MessageType>
<ComplexElementType id="Menu" >
  <description>Available menu items</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <elements>
    <SimpleElementTypeRef idref="MenuItems"/ >
  </elements>
</ComplexElementType>

```

B.3.7 OrganisationTemplate

Atributos: id

Elementos: name, abbreviation, state, dateLaMu, userLaMu

Referencias: contactPerson, template

La organización que participa en el proyecto iniciando o ejecutando TransactionTemplate (transacción).

EJEMPLO Ejemplo al nivel de mensaje:

```

<StandardOrganisationType id="TNO" >
  <name>Netherlands organisation for Applied Scientific Research</name>
  <abbreviation>TNO</abbreviation>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <contactPerson>

```

```

    <StandardPersonType id="..." >
      ...
    </StandardPersonType>
  </contactPerson>
</StandardOrganisationType>

```

Parte asociada al marco de interacción

```

<PersonType id="StandardPersonType" >
  <description>Standard person type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</PersonType>
<OrganisationType id="StandardOrganisationType" >
  <description>Standard organisation type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</OrganisationType>

```

B.3.8 PersonInRole

Atributos: id

Elementos: state, dateLaMu, userLaMu

Referencias: organization, contactPerson, role, substituting, successor

Una persona que cumple un papel particular para una organización determinada.

EJEMPLO Ejemplo al nivel de mensaje:

```

<PersonInRole id="JohnAsCustomer" >
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <contactPerson>
    <StandardPersonType id="..." >
      ...
    </StandardPersonType>
  </contactPerson>
  <organisation>
    <StandardOrganisationType id="..." >
      ...
    </StandardOrganisationType>
  </organisation>
  <role>
    <Consumer idref="..." >
      ...
    </Consumer>
  </role>
</PersonInRole>

```

```
</role>
</PersonInRole>
```

Parte asociada al marco de interacción

```
<PersonType id="StandardPersonType" >
  <description>Standard person type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</PersonType>
<OrganisationType id="StandardOrganisationType" >
  <description>Standard organisation type</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</OrganisationType>
<RoleType id="Consumer" >
  <description>Consuming person</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</RoleType>
```

B.3.9 PersonTemplate

Atributos: id

Elementos: userName, name, state, dateLaMu, userLaMu

La persona que participa en el proyecto cumpliendo un determinado papel o como persona de contacto de una organización en particular.

EJEMPLO Ejemplo al nivel de mensaje:

```
<StandardPersonType id="PBonsma" >
  <userName>bapa</userName>
  <name>Peter Bonsma</name>
  <state>active</state>
  <dateLaMu>2008-02-04T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</StandardPersonType>
```

Parte asociada al marco de interacción

```
<StandardPersonType id="PBonsma" >
  <userName>bapa</userName>
  <name>Peter Bonsma</name>
```

```
<state>active</state>
<dateLaMu>2008-02-04T00:00:00Z</dateLaMu>
<userLaMu>bapa</userLaMu>
</StandardPersonType>
```

B.3.10 ProjectTemplate

Atributos: id

Elementos: name, description, startDate, endDate, state, dateLaMu, userLaMu

El proyecto a ser integrado en este marco de trabajo.

EJEMPLO Ejemplo a nivel de mensaje

```
<StandardProjectType id="IDM2" >
  <name>The project IDM2</name>
  <description>Formalising of the IDM2 Systematics</description>
  <startDate>2008-02-04T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-02-04T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</StandardProjectType>
```

Parte asociada al marco de interacción

```
<ProjectType id="StandardProjectType" >
  <description>Standard project type</description>
  <startDate>2008-02-04T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-02-04T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</ProjectType>
```

B.3.11 RoleTemplate

Atributos: id

Elementos: name, description, state, dateLaMu, userLaMu

El papel en una organización cumplido por un PersonTemplate (persona).

EJEMPLO Ejemplo al nivel de mensaje:

```
<Consumer id="Customer" >
  <name>Role as customer</name>
  <description>The role as customer</description>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</Consumer>
```

Parte asociada al marco de interacción

```
<RoleType id="Consumer" >
  <description> Consuming person</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>20083-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
</RoleType>
```

B.3.12 TransactionPhaseTemplate

Attributes: id

Elements: name, description, dateReached, state, dateLaMu, userLaMu

References: transaction

La fase actual de una transacción.

EJEMPLO Ejemplo al nivel de mensaje:

```
<WaitForMenu id="tp003" >
  <name>...</name>
  <description>Transaction Phase ...</description>
  <dateReached>2008-02-04T00:00:00Z</dateReached>
  <state>active</state>
  <dateLaMu>2008-02-04T00:00:00]]</dateLaMu>
  <userLaMu>bapa</userLaMu>
</WaitForMenu>
```

Parte asociada al marco de interacción

```
<TransactionType id="ObtainMenuTransaction" >
  <description>The transaction to obtain the proper menu</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <initiator>
    <RoleTypeRef idref="Consumer"/ >
  </initiator>
  <executor>
    <RoleTypeRef idref="Employee"/ >
  </executor>
</TransactionType>
<TransactionPhaseType id="WaitForMenu" >
  <description>Menukaart gevraagd maar nog niet gegeven</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
```

```
<userLaMu>bapa</userLaMu>
</TransactionPhaseType>
```

B.3.13 TransactionTemplate

Atributos: id

Referencias: initiator, executor

La transacción que permite que MessageTemplates (mensajes) se intercambien para ejecutar determinadas tareas.

EJEMPLO Ejemplo al nivel de mensaje:

```
<ObtainMenuTransaction id="TheTransaction" >
  <number>001</number>
  <name>...</name>
  <description>...</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-01-23T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <initiator>
    <PersonInRole id="..." >
      ...
    </PersonInRole>
  </initiator>
  <executor>
    <PersonInRole id="..." >
      ...
    </PersonInRole>
  </executor>
</ObtainMenuTransaction>
```

Parte asociada al marco de interacción

```
<TransactionType id="ObtainMenuTransaction" >
  <description>The transaction to obtain the proper menu</description>
  <startDate>2008-01-23T00:00:00Z</startDate>
  <endDate>2008-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
  <userLaMu>bapa</userLaMu>
  <initiator>
    <RoleTypeRef idref="Consumer"/ >
  </initiator>
  <executor>
    <RoleTypeRef idref="Employee"/ >
  </executor>
</TransactionType>
```

B.4 Attributes

B.4.1 id

Código único para identificar un ejemplar de objeto dentro de un mensaje.

Nota: El valor ID debería ser único dentro del documento XML. En consecuencia, un número de serie sencillo ya sería suficiente para satisfacer esta condición. Como resultado, no hay ninguna razón convincente para limitar los valores de ID a GUIDs (aunque la mayoría de las implementaciones aplican GUID). Marque que el estándar XML (<http://www.w3.org/XML/>) requiere que un ID comience con una letra o “_”.

B.5 Elements

B.5.1 Abbreviation

Abreviatura del nombre de la organización. Este elemento se utilizará como prefijo del número de transacción para identificación de la transacción.

B.5.2 category

Categoría para clasificar este ejemplar de objeto.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <category>...</category>
  ...
</...>
```

B.5.3 creationDate

Fecha de creación de un ejemplar de objeto específico en este mensaje.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <category>...</category>
  ...
</...>
```

B.5.4 dateLaMu

Fecha de la última mutación.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <creationDate>2007-12-03T00:00:00Z</creationDate>
```

```
<...>
```

B.5.5 dateReached

Fecha “dateReached” es un atributo de fecha y hora de “TransactionPhaseTemplate”. Es la fecha y la hora en la que se alcanzó una fase particular.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <dateReached>2008-02-04T00:00:00Z</dateReached>
  ...
</...>
```

B.5.6 dateRead

Fecha de lectura de este mensaje.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <dateRead>2007-12-03T00:00:00Z</dateRead>
  ...
</...>
```

B.5.7 dateSend

Fecha de envío de este mensaje.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <dateSend>2007-12-03T00:00:00Z</dateSend>
  ...
</...>
```

B.5.8 description

Description of this object instance.

EXAMPLE

Example at message level:

```
<Projectexecutor id="TNO" >
  ...
  <description>...</description>
  ...
</Projectexecutor>
```

B.5.9 documentIdentification

Número único o referencia de un documento o archivo para identificar el documento.

Nota: Esta parte de la Norma ISO 29481 ofrece la libertad de distinguir entre versiones de documentos y versiones de archivos (véase B.5.13).

B.5.10 documentReference

Referencia para identificar un archivo o documento.

B.5.11 documentVersion

Versión de un documento o archivo.

B.5.12 endDate

Fecha de caducidad de un ejemplar de objeto específico en este mensaje.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
...
  <endDate>2007-12-03</endDate>
...
</...>
```

B.5.13 fileLocation

Ubicación del archivo, preferiblemente en una dirección de Internet o una ruta compartida de servidor.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
...
  <fileLocation>\\srv-path\Public\project-x\docs</fileLocation>
...
</...>
```

Nota: Esta parte de la Norma ISO 29481 ofrece la libertad de distinguir entre versiones de documentos y versiones de archivos (véase B.5.9).

B.5.14 fileType

Tipo de archivo, preferiblemente de tipo MIME (Multipurpose Internet Mail Extensions).

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
...
  <fileType>text/plain</fileType>
```

```
<...>
</...>
```

B.5.15 fileVersion

Versión del archivo, un número entero creciente o la fecha.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <fileVersion>20071202</fileVersion>
  ...
</...>
```

B.5.16 identification

Identificación de este mensaje orientada a humanos.

EJEMPLOS Ejemplo al nivel de mensaje:

```
<AssignmentConfirmation id="X" >
  <identification>This message contains the assignment X related to part
  Y</identification>
  ...
</AssignmentConfirmation>
```

B.5.17 initiatingTransactionMessageID

Referencia al mensaje perteneciente a una transacción secundaria que inició este mensaje.

B.5.18 initiatorToExecutor

La dirección de comunicación de este mensaje específico.

EJEMPLO Ejemplo al nivel de mensaje:

```
<ExampleMessage id="..." >
  ...
  <initiatorToExecutor>false</initiatorToExecutor>
  ...
  <transaction>
    <ExampleTransaction id="..." >
      ...
      <initiator>
        <PersonInRole id="A" >
          ...
        </PersonInRole>
      </initiator>
      <executor>
        <PersonInRole id="B" >
          ...
        </PersonInRole>
      </executor>
    </ExampleTransaction>
  </transaction>
</ExampleMessage>
```

```
        </PersonInRole>
      </executor>
    </ExampleTransaction>
  </transaction>
  ...
</ExampleMessage>
```

En este ejemplo, *ExampleMessage* viajará desde B (ejecutor) a A (iniciador).

B.5.19 language

El lenguaje de contenido del archivo adjunto.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <language>EN</language>
  ...
</...>
```

B.5.20 name

El nombre (STRING).

B.5.21 number

Número de transacción.

Nota: Número de elemento. El número de secuencia de una transacción (véase también el elemento anterior).

B.5.22 objectCode

Posibilidad de relacionarse con un índice externo. Por ejemplo, una estructura de desglose de trabajo, paquetes de trabajo o las especificaciones del edificio.

B.5.23 result

Resultado de esta transacción.

B.5.24 startDate

Fecha de comienzo de un ejemplar de objeto en este mensaje.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <startDate>2007-12-03</startDate>
  ...
</...>
```

B.5.25 state

Estado actual de este ejemplar de objeto.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <state>active</state>
  ...
</...>
```

B.5.26 userLaMu

Usuario que realizó la última modificación (MINS: no se trata de una ejemplarización PersonTypeInstance).

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <userLaMu>Peter Bonsma</userLaMu>
  ...
</...>
```

B.5.27 userName

Iniciales del usuario, por ejemplo, un conjunto compuesto por tres letras.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <userName>BAP</userName>
  ...
</...>
```

B.5.28 versionNo

Número de versión del ejemplar de objeto conectado a este proyecto. Después de una modificación de este ejemplar de objeto, este campo también debería actualizarse.

EJEMPLO Ejemplo al nivel de mensaje:

```
<... id="..." >
  ...
  <versionNo>23</versionNo>
  ...
</...>
```

B.6 References

B.6.1 appendixGroup

Una subdivisión para identificar un cierto archivo adjunto.

EJEMPLO Ejemplo al nivel de mensaje:

```
<Appendix id="..." >
  ...
  <appendixGroup>
    <AppendixGroup id="..." >
      ...
    </AppendixGroup>
  </appendixGroup>
  ...
</Appendix>
```

Se considera que el marco asociado ha definido un AppendixType *Appendix*.

B.6.2 contactPerson

La persona conectada a un objeto PersonInRole o conectada a una organización específica.

EJEMPLO Ejemplo al nivel de mensaje:

```
<Organisation id="..." >
  ...
  <contactPerson>
    <Person id="..." >
      ...
    </Person>
  </contactPerson>
</Organisation>
```

Se considera que el marco asociado ha definido un OrganisationType *Organization* y un PersonType *Person*.

B.6.3 executor

La PersonInRole que hará de ejecutor de esta transacción.

B.6.4 group

El grupo general para montar un conjunto de archivos adjuntos.

EJEMPLO Ejemplo al nivel de mensaje:

```
<AppendixGroup id="..." >
  ...
  <group>
```

```

    <Group id="..." >
      ...
    </Group>
  </group>
  ...
</AppendixGroup>

```

Se considera que el marco asociado ha definido un GroupType *Group*.

B.6.5 initiator

La PersonInRole que hará de iniciador de esta transacción.

B.6.6 message

El mensaje que contiene un archivo adjunto específico.

EJEMPLO Ejemplo al nivel de mensaje:

```

<Appendix id="..." >
  ...
  <message>
    <Message id="..." >
      ...
    </Message>
  </message>
  ...
</Appendix>

```

Se considera que el marco asociado ha definido un AppendixType *Appendix* y un MessageType *Message*.

B.6.7 messageInTransaction

Referencia a la posición de este mensaje en el flujo de la transacción.

B.6.8 organization

La organización perteneciente a un objeto PersonInRole.

EJEMPLO Ejemplo al nivel de mensaje:

```

<PersonInRole id="..." >
  ...
  <organisation>
    <Organisation id="..." >
      ...
    </Organisation>
  </organisation>
  ...
</PersonInRole>

```

Se considera que el marco asociado ha definido un OrganisationType *Organization*.

B.6.9 role

Referencia a un papel de una organización, cumplido por un PersonTemplate (persona).

B.6.10 substituting

PersonInRole que en nombre de esta persona está autorizado a enviar mensajes.

Nota: Referencia substituting. Una referencia formal de la PersonInRole de esta transacción para habilitar a otra persona a actuar en ausencia de la PersonInRole formal que cumple esta función. Ambos deberían referirse al mismo tipo de rol.

B.6.11 sucesor

Sucesor de otra persona en un determinado papel.

B.6.12 transaction

La transacción que contiene un grupo específico, un mensaje o una fase de transacción.

EJEMPLO Ejemplo al nivel de mensaje:

```
<Message id="..." >
  ...
  <transaction>
    <Transaction id="..." >
      ...
    </Transaction>
  </transaction>
  ...
</Message>
```

Se considera que el marco asociado ha definido un MessageType *Message* y un TransactionType *Transaction*.

Anexo C (Informativo)

Ejemplo de mapa de interacción simplificado de una oficina de diseño

C.1 Generalidades

En nuestro ejemplo de creación de un marco de interacción, el alcance de la interacción se ha simplificado en el seno de una oficina de diseño (véase el ejemplo en capítulo 4). La interacción se define en el marco de la declaración de:

- Funciones,
- Transacciones,
- Mensajes en transacciones,
- El orden de los mensajes en la transacción,
- Elementos de datos en los mensajes.

C.2 Funciones y transacciones

Para ilustrar el alcance de la interacción en nuestro ejemplo, las acciones comunicativas se representan en un “mapa de interacción”, que muestra los roles vinculados por las transacciones.

Roles:

- CR₁: Cliente;
- R₁: Jefe de proyecto;
- R₂: Ingeniería de sistemas;
- R₃: Ingeniería 3D;
- R₄: Ingeniería de costes.

Las interacciones se pueden resumir en una tabla de rol de transacción.

Tabla C.1 – Tabla de rol de transacción de una oficina de diseño simplificado

Tipo de transacción	Rol de iniciador	Rol de ejecutor
T ₁ Entrega diseño	CR ₁ : Cliente	R ₁ : Jefe de proyecto
T ₂ Entrega de las especificaciones del sistema	R ₁ : Jefe de proyecto	R ₂ : Ingeniería de sistemas
T ₃ Entrega modelo 3D	R ₁ : Jefe de proyecto	R ₃ : Ingeniería 3D
T ₄ Cálculo del coste de entrega	R ₁ : Jefe de proyecto	R ₄ : Ingeniería de costes

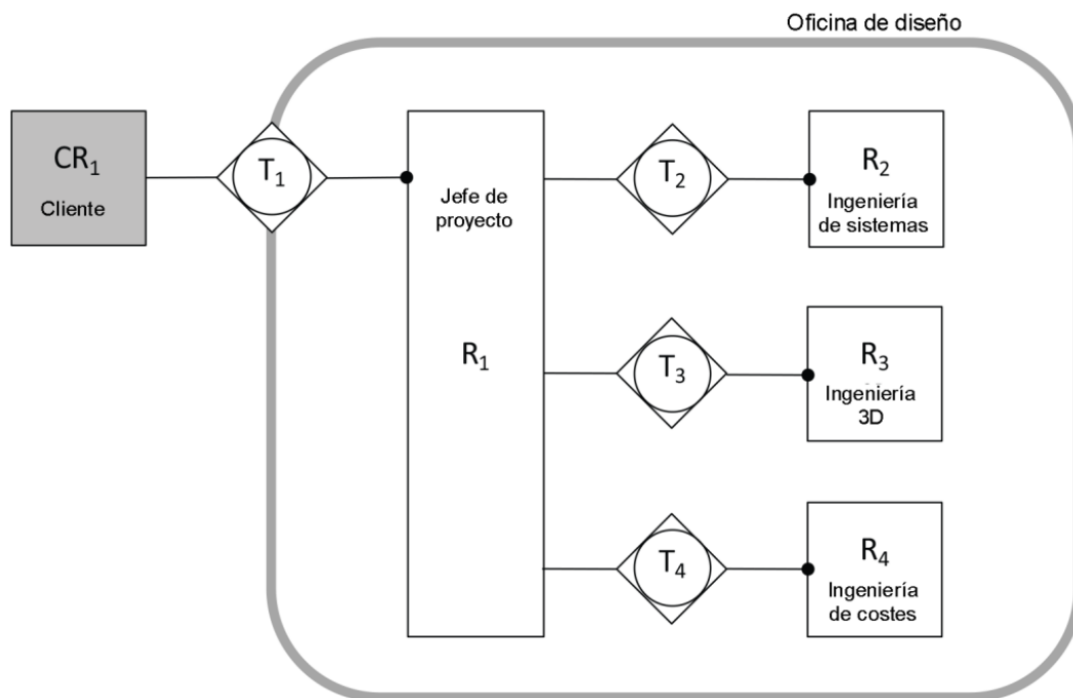


Figura C.1 – Mapa de interacción, identificando los tipos de rol y tipos de transacciones relevantes}

C.2.1 RoleType

Los roles se definen como 'RoleTypes' en el marco de trabajo. Cada RoleType se puede definir en términos de:

- Un ID (identificación de roleType),
 - [Debería ser un valor de ID XML válidos (nombre calificado). Hay ciertas restricciones: *Único dentro del modelo, *No se permiten ciertos caracteres ('/', '\$', '#', '@',...), *Sin espacios, *No se debería empezar con un dígito]
- Descripción (especificación de roleType),
- Responsabilidades (alcance, tarea, tarea de apoyo y retroalimentación),
- Metadatos (última mutación, información de ayuda, etc.)

Los cuatro roles de nuestro ejemplo se han definido en el marco de trabajo (véase el ejemplo siguiente en XML):

```

<RoleType id="CR1_Client" >
  <description>CR1: Client</description>
  <startDate>2010-10-29T00:00:00.000+02:00</startDate>
  <endDate>2099-10-29T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>

```

```

    <helpInfo></helpInfo>
    <code></code>
    <responsibilityScope>Is responsible for the realisation of the
project in terms of time, quality and
    cost < /responsibilityScope>
    <responsibilityTask></responsibilityTask>
    <responsibilitySupportTask></responsibilitySupportTask>
    <responsibilityFeedback></responsibilityFeedback>
  </RoleType>
  <RoleType id="R1_Project_Leading" >
    <description>R1: Project Leading</description>
    <startDate>2010-10-29T00:00:00.000+02:00</startDate>
    <endDate>2099-10-29T00:00:00.000+02:00</endDate>
    <state>active</state>
    <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code></code>
    <responsibilityScope>Has to deliver the project according to the
contract
    requirements</responsibilityScope>
    <responsibilityTask></responsibilityTask>
    <responsibilitySupportTask></responsibilitySupportTask>
    <responsibilityFeedback></responsibilityFeedback>
  </RoleType>
  <RoleType id="R2_System_Engineering" >
    <description>R2: System Engineering</description>
    <startDate>2010-10-29T00:00:00.000+02:00</startDate>
    <endDate>2099-10-29T00:00:00.000+02:00</endDate>
    <state>active</state>
    <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code></code>
    <responsibilityScope>Has to deliver a system specification according
to contractual
    agreements</responsibilityScope>
    <responsibilityTask></responsibilityTask>
    <responsibilitySupportTask></responsibilitySupportTask>
    <responsibilityFeedback></responsibilityFeedback>
  </RoleType>
  <RoleType id="R3_3D_Engineering" >
    <description>R3: 3D Engineering</description>
    <startDate>2010-10-29T00:00:00.000+02:00</startDate>
    <endDate>2099-10-29T00:00:00.000+02:00</endDate>
    <state>active</state>
    <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code></code>

```

```

    <responsibilityScope>Has to deliver a 3D model according to
contractual
    agreements</responsibilityScope>
    <responsibilityTask></responsibilityTask>
    <responsibilitySupportTask></responsibilitySupportTask>
    <responsibilityFeedback></responsibilityFeedback>
</RoleType>
<RoleType id="R4_Cost_Engineering" >
    <description>R4: Cost Engineering</description>
    <startDate>2010-10-29T00:00:00.000+02:00</startDate>
    <endDate>2099-10-29T00:00:00.000+02:00</endDate>
    <state>active</state>
    <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code></code>
    <responsibilityScope>Has to deliver a cost calculation according to
contractual
    agreements</responsibilityScope>
    <responsibilityTask></responsibilityTask>
    <responsibilitySupportTask></responsibilitySupportTask>
    <responsibilityFeedback></responsibilityFeedback>
</RoleType>

```

C.3 Mensaje en una transacción

Una transacción contiene un conjunto de mensajes que se intercambian para un propósito determinado. La transacción también estipula los papeles participantes, el punto en el ciclo de vida, y la secuencia en la que los mensajes deberían entregarse (si procediera).

C.3.1 Tipo de transacción

Las transacciones se definen como 'TransactionTypes' en el marco de trabajo. Cada TransactionType se puede definir en términos de:

- Un ID (identificación de TransactionType),
- Descripción (especificación de TransactionType),
- Metadatos (resultado de TransactionType, helpinfo, etc.),
- Los RoleTypes que están involucrados en el TransactionType.

Los cuatro TransactionTypes en nuestro ejemplo se representan a continuación en una vista XML:

- T1: Entrega del diseño;
- T2: Entrega de la especificación del sistema;
- T3: Entrega del modelo 3D;
- T4: Entrega del cálculo de costes.

```
<TransactionType id="T1" >
```

```

    <description>T1 Deliver design </description>
    <startDate>2010-10-29T00:00:00.000+02:00</startDate>
    <endDate>2099-10-29T00:00:00.000+02:00</endDate>
    <state>active</state>
    <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code></code>
    <result>Design is delivered</result>
    <initiator>
      <RoleTypeRef idref="CR1_Client"/ >
    </initiator>
    <executor>
      <RoleTypeRef idref="R1_Project_Leading"/ >
    </executor>
  </TransactionType>
  <TransactionType id="T2" >
    <description>T2 Deliver system specification</description>
    <startDate>2010-10-29T00:00:00.000+02:00</startDate>
    <endDate>2099-10-29T00:00:00.000+02:00</endDate>
    <state>active</state>
    <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code></code>
    <result>System specification is delivered</result>
    <initiator>
      <RoleTypeRef idref="R1_Project_Leading"/ >
    </initiator>
    <executor>
      <RoleTypeRef idref="R2_System_Engineering"/ >
    </executor>
  </TransactionType>
  <TransactionType id="T3" >
    <description>T3 Deliver 3D model</description>
    <startDate>2010-10-29T00:00:00.000+02:00</startDate>
    <endDate>2099-10-29T00:00:00.000+02:00</endDate>
    <state>active</state>
    <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code></code>
    <result>3D model is delivered</result>
    <initiator>
      <RoleTypeRef idref="R1_Project_Leading"/ >
    </initiator>
    <executor>
      <RoleTypeRef idref="R3_3D_Engineering"/ >
    </executor>
  </TransactionType>
  <TransactionType id="T4" >

```

```

<description>T4 Deliver cost calculation</description>
<startDate>2010-10-29T00:00:00.000+02:00</startDate>
<endDate>2099-10-29T00:00:00.000+02:00</endDate>
<state>active</state>
<dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
<userLaMu>Ronald</userLaMu>
<language></language>
<category></category>
<helpInfo></helpInfo>
<code></code>
<result>Cost calculation is delivered</result>
<initiator>
  <RoleTypeRef idref="R1_Project_Leading"/ >
</initiator>
<executor>
  <RoleTypeRef idref="R4_Cost_Engineering"/ >
</executor>
</TransactionType>

```

IDM establece una distinción entre un rol que hace una petición (el iniciador) y el rol al que da efecto a esa petición (el ejecutor). Una transacción sólo tendrá un rol iniciador y un solo rol ejecutante. Por ejemplo, TransactionType 'T3 Deliver 3D model' solo puede ser iniciado por el rol R1 Jefe de proyecto y ejecutado por el rol R3, que es la ingeniería 3D.

Mediante el uso de transacciones, la transferencia de información se introduce en un contexto procedimental.

C.3.2 TransactionPhaseType

Para indicar el estado de interacción dentro de una transacción, las fases pueden estar relacionadas con los TransactionTypes.

De hecho, los TransactionPhaseTypes dentro de TransactionTypes representan el patrón de transacción. Las definiciones TransactionPhaseType proporcionan una identificación de TransactionPhaseType y algunos metadatos. El número de fases y su descripción no están restringidos.

En nuestro ejemplo, se han definido los siguientes TransactionPhaseTypes, que se refieren a la figura 2 del capítulo 3:

- Resultado deseado;
- Resultado solicitado;
- Resultado prometido;
- Resultado producido;
- Resultado declarado;
- Resultado aceptado.

El ejemplo siguiente muestra al definición de TransactionPhaseType en XML:

```
<TransactionPhaseType id="Desired_Result" >
  <description>Desired result</description>
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <code>-</code>
</TransactionPhaseType>
<TransactionPhaseType id="Result_Accepted" >
  <description>Result accepted</description>
  <startDate>2010-11-04T00:00:00.000+01:00</startDate>
  <endDate>2099-11-04T00:00:00.000+01:00</endDate>
  <state>active</state>
  <dateLaMu>2010-11-04T00:00:00.000+01:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <code>-</code>
</TransactionPhaseType>
<TransactionPhaseType id="Result_Produced" >
  <description>Result produced</description>
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <code>-</code>
</TransactionPhaseType>
<TransactionPhaseType id="Result_Promised" >
  <description>Result promised</description>
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <code>-</code>
</TransactionPhaseType>
<TransactionPhaseType id="Result_Requested" >
  <description>Result requested</description>
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
```

```

    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code>-</code>
</TransactionPhaseType>
<TransactionPhaseType id="Result Stated" >
    <description>Result stated</description>
    <startDate>2010-11-04T00:00:00.000+01:00</startDate>
    <endDate>2099-11-04T00:00:00.000+01:00</endDate>
    <state>active</state>
    <dateLaMu>2010-11-04T00:00:00.000+01:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code>-</code>
</TransactionPhaseType>

```

C.3.3 MessageType

Un mensaje es un modelo de información que contiene datos. El MessageType define la estructura y proporciona los elementos para el contenido del mensaje.

Dentro de cada mensaje, hay grupos de segmentos (elementos compuestos). Estos elementos compuestos son definidos por ComplexElementTypes. Dentro de cada segmento, puede haber una mezcla de elementos compuestos y/o elementos singulares. Los elementos singulares son definidos por SimpleElementTypes.

El ejemplo siguiente muestra en XML la definición de los MessageTypes que se representan gráficamente en la figura 5 del capítulo 3:

- Solicitud de modelo 3D;
- Trabajo realizado y solicitud de aprobación;
- Solicitar ajustes;
- Trabajos aprobados;
- Trabajo no aprobado.

```

<MessageType id="Request_for_3D_model" >
    <description>Request for 3D model</description>
    <startDate>2010-11-04T00:00:00.000+01:00</startDate>
    <endDate>2099-11-04T00:00:00.000+01:00</endDate>
    <state>active</state>
    <dateLaMu>2010-11-04T00:00:00.000+01:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code>-</code>
    <complexElements>
        <ComplexElementTypeRef idref="ResultRequestedComplexElement" / >
        <ComplexElementTypeRef idref="WorkIdentificationComplexElement" / >
        <ComplexElementTypeRef idref="WorkSpecificationComplexElement" / >
    </complexElements>
</MessageType>

```

```

        <ComplexElementTypeRef idref="RemarksComplexElement"/ >
        <ComplexElementTypeRef idref="DeadlineComplexElement"/ >
    </complexElements>
</MessageType>
<MessageType id="Work_done_and_request_approval" >
    <description>Work done and request approval</description>
    <startDate>2010-11-04T00:00:00.000+01:00</startDate>
    <endDate>2099-11-04T00:00:00.000+01:00</endDate>
    <state>active</state>
    <dateLaMu>2010-11-04T00:00:00.000+01:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code>-</code>
    <complexElements>
        <ComplexElementTypeRef idref="RequestApprovalComplexElement"/ >
        <ComplexElementTypeRef idref="RemarksComplexElement"/ >
        <ComplexElementTypeRef idref="DeadlineComplexElement"/ >
        <ComplexElementTypeRef idref="WorkIdentificationComplexElement"/ >
        <ComplexElementTypeRef idref="WorkSpecificationComplexElement"/ >
    </complexElements>
</MessageType>
<MessageType id="Request_adjustments" >
    <description>Request adjustments</description>
    <startDate>2010-11-04T00:00:00.000+01:00</startDate>
    <endDate>2099-11-04T00:00:00.000+01:00</endDate>
    <state>active</state>
    <dateLaMu>2010-11-04T00:00:00.000+01:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code>-</code>
    <complexElements>
        <ComplexElementTypeRef idref="RequestAdjustmentsComplexElement"/ >
        <ComplexElementTypeRef idref="RemarksComplexElement"/ >
        <ComplexElementTypeRef idref="DeadlineComplexElement"/ >
        <ComplexElementTypeRef idref="WorkIdentificationComplexElement"/ >
        <ComplexElementTypeRef idref="WorkSpecificationComplexElement"/ >
    </complexElements>
</MessageType>
<MessageType id="Work_approved" >
    <description>Work approved</description>
    <startDate>2010-11-04T00:00:00.000+01:00</startDate>
    <endDate>2099-11-04T00:00:00.000+01:00</endDate>
    <state>active</state>
    <dateLaMu>2010-11-04T00:00:00.000+01:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code>-</code>
    <complexElements>
        <ComplexElementTypeRef idref="ApprovalComplexElement"/ >
        <ComplexElementTypeRef idref="RemarksComplexElement"/ >
        <ComplexElementTypeRef idref="WorkIdentificationComplexElement"/ >

```

```

        <ComplexElementTypeRef idref="WorkSpecificationComplexElement"/ >
    </complexElements>
</MessageType>
<MessageType id="Work_not_approved" >
    <description>Work not approved</description>
    <startDate>2010-11-04T00:00:00.000+01:00</startDate>
    <endDate>2099-11-04T00:00:00.000+01:00</endDate>
    <state>active</state>
    <dateLaMu>2010-11-04T00:00:00.000+01:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language></language>
    <category></category>
    <helpInfo></helpInfo>
    <code>-</code>
    <complexElements>
        <ComplexElementTypeRef idref="DisapprovalComplexElement"/ >
        <ComplexElementTypeRef idref="RemarksComplexElement"/ >
        <ComplexElementTypeRef idref="WorkIdentificationComplexElement"/ >
        <ComplexElementTypeRef idref="WorkSpecificationComplexElement"/ >
    </complexElements>
</MessageType>

```

MessageType también se utiliza para identificar mensajes dentro de una transacción. La relación entre un messageType y un transactionType se define en otro Type: el MessageInTransactionType.

C.3.4 MessageInTransactionType

Una transacción contiene un conjunto de mensajes que se intercambian para un propósito particular. Las definiciones MessageInTransactionType proporcionan la vinculación de MessageTypes a TransactionTypes. Un MessageType idéntico se puede vincular a todos los TransactionTypes definidos. También es posible vincular un MessageType idéntico más de una vez (incluso ilimitado) con el mismo TransactionType.

Cada MessageInTransactionType se puede definir en términos de

- Un ID (identificación del MessageInTransactionType),
- La dirección en la que se debería enviar el mensaje (Del Initiator al Executor = true, del Executor al Initiator = Falso),
- Metadatos de MessageInTransactionType,
- La referencia al MessageType (cuya posición se define en MessageInTransactionType),
- La referencia a los TransactionType (s) para enlazar los referidos MessageTypes,
- La referencia a la TransactionPhase en qué estado se lleva el proceso (después de enviar el MessageType dentro de TransactionType como ha sido definido por el MessageInTransactionType),
- La referencia del GroupType

El ejemplo siguiente muestra en XML la definición de dos MessageInTransactionTypes

en nuestro ejemplo de entrega de un modelo 3D. El ejemplo se puede entender haciendo referencia a la figura 5 (en el proyecto de Norma ISO/WD 29481-2 IDM parte 2 Mapa de interacción; capítulo 3).

El ejemplo XML expuesto a continuación ilustra que el TransactionType T3 comienza con MessageType 'Solicitud de modelo 3D'. Es el primer mensaje dentro de la transacción porque no hay MessageInTransactionType definido en MessageInTransactionType 'MessageTransaction1'. La dirección de MessageInTransaction1 se señala de Iniciador a Exexutor (value = true).

El segundo MessageInTransactionType definido por 'MessageInTransaction2' hace referencia a MessageType 'Trabajo realizado y solicitud de aprobación'. La figura 5 muestra que este mensaje debería seguir dos mensajes anteriores:

- Solicitud de modelo 3D;
- Solicitar ajustes.

Estos dos mensajes anteriores se han definido como MessageInTransactionTypes (nº 1 y 3) dentro de El MessageInTransactionType 2. La dirección de MessageInTransaction1 se señala desde Executor hasta Iniciador (valor = falso).

```
<MessageInTransactionType id="MessageInTransactie1" >
  <requiredNotify>0</requiredNotify>
  <dateLaMu>2010-11-05T00:00:00.000+01:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <received>0</received>
  <send>0</send>
  <state></state>
  <initiatorToExecutor>true</initiatorToExecutor>
  <message>
    <MessageTypeRef idref="Request_for_3D_model"/ >
  </message>
</transaction>
  <TransactionTypeRef idref="T3"/ >
</transaction>
<transactionPhase>
  <TransactionPhaseTypeRef idref="Result_Requested"/ >
</transactionPhase>
<group>
  <GroupTypeRef idref="StandardGroup"/ >
</group>
</MessageInTransactionType>
<MessageInTransactionType id="MessageInTransactie2" >
  <requiredNotify>0</requiredNotify>
  <dateLaMu>2010-11-05T00:00:00.000+01:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <received>0</received>
  <send>0</send>
  <state></state>
  <initiatorToExecutor>>false</initiatorToExecutor>
  <message>
    <MessageTypeRef idref="Work_done_and_request_approval"/ >
  </message>
```

```

<previous>
  <MessageInTransactionTypeRef idref="MessageInTransactie1"/ >
  <MessageInTransactionTypeRef idref="MessageInTransactie3"/ >
</previous>
<transaction>
  <TransactionTypeRef idref="T3"/ >
</transaction>
<transactionPhase>
  <TransactionPhaseTypeRef idref="Result_Produced"/ >
</transactionPhase>
<group>
  <GroupTypeRef idref="StandardGroup"/ >
</group>
</MessageInTransactionType>

```

C.3.5 AppendixTypes

Los archivos adjuntos pueden estar vinculados a mensajes. Un requisito de intercambio se puede transferir como archivo adjunto al papel de ejecutor y el resultado (contribución al BIM) se entrega al iniciador.

Las definiciones de AppendixType proporcionan restricción de información de elementos de los elementos relacionados con documentos mediante la adición de metadatos a documentos adjuntos a un mensaje.

En el listado en XML siguiente se muestra la definición del AppendixType. Los metadatos se definen por SimpleElementTypes en el ComplexElementType 'AppendixComplexElement'.

```

<AppendixType id="StandardAppendixType" >
  <description>Standard Appendix</description>
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <complexElements>
    <ComplexElementTypeRef idref="AppendixComplexElement"/ >
  </complexElements>
</AppendixType>

```

C.3.6 Limitando el contenido de los mensajes

El tercer y último paso para definir el marco de interacción es definir los componentes dentro de MessageTypes con el fin de:

- Estructurar los MessageTypes;
- Restringir la entrada de contenido.

La definición de MessageType ya contiene los elementos compuestos de un mensaje.

Estos elementos compuestos se definen como `ComplexElementTypes`.

Los `ComplexElementTypes` consisten en `SimpleElementTypes` (también son posibles otros `ComplexElementTypes`). Los `SimpleElementTypes` están relacionados con `UserDefinedTypes` para definir restricciones a la entrada de contenido. Esta estructura se representa gráficamente a continuación.

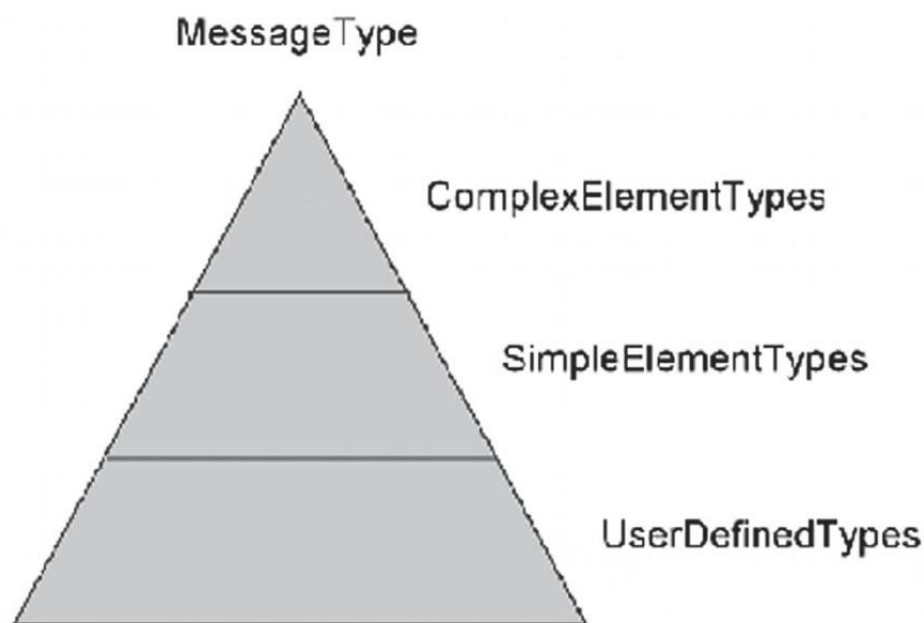


Figura C.2 – Componentes del modelado de un mensaje

C.3.7 `ComplexElementTypes`

Las definiciones de `ComplexElementType` proporcionan:

- Elementos de información de elementos de restricción;
- Utilizando los mecanismos de la jerarquía de definición de tipos (en XML) para derivar un tipo complejo de otro tipo simple o complejo,
- Especificación de las contribuciones infoset de validación post-esquema para elementos,
- Limitar la capacidad de derivar tipos adicionales de un tipo complejo dado.

Refiriéndonos a nuestro ejemplo de `MessageType` 'Solicitud de un modelo 3D', relacionado con la transacción T3 (Entrega del modelo 3D), este `MessageType` contiene los siguientes `ComplexElementTypes`:

- `resultRequestComplexElement`;
- `workIdentificationComplexElement`;
- `workSpecificationComplexElement`;
- `remarksComplexElement`;

- deadlineComplexElement.

El ComplexElementType 'WorkSpecification' se Muestra a continuación en XML. Un ComplexElementType define su identificación, la referencia a los elementos simples que dan contenido al complexElementType y a sus metadatos.

El listado XML de ComplexElementType 'WorkSpecificationComplexElement' muestra que ComplexElementType contiene dos tipos SimpleElementTypes:

- ScopeOfWork;
- CostEstimation.

```
<ComplexElementType id="WorkSpecificationComplexElement" >
  <description>Work specification</description>
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <simpleElements>
    <SimpleElementTypeRef idref="ScopeOfWork"/ >
    <SimpleElementTypeRef idref="CostEstimation"/ >
  </simpleElements>
</ComplexElementType>
```

C.3.9 UserDefinedTypes

Los UserDefinedTypes se utilizan para definir las restricciones a las entradas de contenido (definiendo dataTypes). A todas las posibles restricciones XML se les permite definir los UserDefinedTypes dentro de un marco de interacción. En nuestro ejemplo, sólo se han utilizado el tipo de datos 'STRING' Y 'EURO'.

El ejemplo siguiente muestra en el listado XML la definición de un UserDefinedType:

- Identificación,
 - Metadatos,
 - baseType,
 - restricción XSD (para definir una restricción más específica o definir tipos de enumeración).
- ```
- <UserDefinedType id="EURO" >
 <description>Euro</description>
 <state>active</state>
 <dateLaMu>2010-12-10T12:35:19.485+01:00</dateLaMu>
 <userLaMu>Ronald</userLaMu>
 <baseType>DECIMAL</baseType>
 <xsdRestriction><xs:fractionDigits value="2"/ > </xsdRestriction >
 <helpInfo>amounts in Euro, specified in two decimals</helpInfo>
</UserDefinedType>
```

```
- <UserDefinedType id="STRING" >
 <description>Character strings</description>
 <state>active</state>
 <dateLaMu>2010-10-01T00:00:00</dateLaMu>
 <userLaMu>Ronald</userLaMu>
 <baseType>STRING</baseType>
 <xsdRestriction />
 <language />
 <helpInfo>The string datatype represents character strings in XML. The
'value space' of string is the set of finite-length sequences of
characters.</helpInfo>
</UserDefinedType>
```

### C.3.10 Finalización

Todos los ElementTypes han sido definidos para completar el marco de interacción. Con el fin de verificar la corrección, el marco debe validarse contra su esquema XSD. El siguiente paso es promover el marco de interacción para obtener un esquema de interacción. Estas acciones se han descrito en el apartado 4.8.

## Anexo D (Informativo)

### Principios para el algoritmo Promotor

#### D.1 Principios para el algoritmo Promotor

Un Promotor es un algoritmo que convierte un marco de interacción en un esquema de interacción. Dado que un marco de interacción se registra en XML y un esquema de interacción en XSD, se puede concluir que un Promotor extrae los elementos de datos del marco de interacción en la estructura entidad-relación de un esquema de interacción.

El algoritmo básico de un Promotor eleva todos los elementos de datos “\* Type” del marco de interacción (RoleType, TransactionType, MessageType, ComplexElementType, SimpleElementType, etc.) en elementos tipo XSD complejos. Por supuesto, han de cumplirse ciertas reglas para garantizar que el XSD resultante sea un esquema XML significativo. Por ejemplo, el valor del atributo ID de un elemento de tipo \*Type se interpreta como el nombre de su equivalente de elementos complejos XSD. Como resultado, el atributo ID en un marco de interacción no debería ser del tipo ID = “Role003” sino más bien como ID = “Project\_Manager”.

El XSD está organizado de tal manera que los MessageTypes son los elementos complejos primarios para estructurar el contenido de un mensaje a nivel del intercambio real en una transacción. Dicho mensaje debería contener suficiente información para rastrear la posición real de este mensaje en el flujo total de mensajes y para determinar qué mensajes sucesivos pueden ser enviados desde esta posición.

Para generar los tipos de atributos y tipos de relación correctos para el XSD resultante, el Promotor consulta un archivo que describe qué plantilla se debería utilizar para cada tipo de elemento complejo (archivo de plantillas).

**Bibliografía**

- [1] Dietz J.L.G. (2006). *Enterprise Ontology: Theory and Methodology*, Springer; 1 edition, 240 pages, ISBN: 3540291695