ГОСТ Р 10.0.04-2019/ИСО 29481-2:2012

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Система стандартов информационного моделирования зданий и сооружений
ИНФОРМАЦИОННОЕ МОДЕЛИРОВАНИЕ В СТРОИТЕЛЬСТВЕ
Справочник по обмену информацией
Часть 2
Структура взаимодействия
System of standards on information modeling of buildings and structures. Building information models. Information delivery manual. Part 2. Interaction framework

ОКС 91.010.01
35.240.67
35.240.01

Дата введения 2019-09-01

Предисловие

1 ПОДГОТОВЛЕН Ассоциацией организаций по развитию технологий информационного моделирования в строительстве и ЖКХ (БИМ-Ассоциация) на основе собственного перевода на русский язык англоязычной версии стандарта, указанного в пункте 4

2 ВНЕСЕН Проектным техническим комитетом по стандартизации ПТК 705 "Технологии информационного моделирования на всех этапах жизненного цикла объектов капитального строительства и недвижимости"

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 5 июня 2019 г. N 280-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 29481-2:2012* "Информационное моделирование в строительстве. Справочник по обмену информацией. Часть 2. Структура взаимодействия" (ISO 29481-2:2012 "Building information models - Information delivery manual - Part 2: Interaction framework", IDT).

_____

* Доступ к международным и зарубежным документам, упомянутым в тексте, можно получить, обратившись в Службу поддержки пользователей. - Примечание изготовителя базы данных.

Наименование настоящего стандарта изменено относительно наименования указанного международного стандарта для приведения в соответствие с ГОСТ Р 1.5-2012 (пункт 3.5).

При применении настоящего стандарта рекомендуется использовать вместо ссылочных международных стандартов соответствующие им национальные стандарты, сведения о которых приведены в дополнительном приложении ДА

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. N 162-ФЗ "О стандартизации в Российской Федерации"*. Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе "Национальные стандарты", а официальный текст изменений и поправок - в ежемесячном информационном указателе "Национальные стандарты". В случае пересмотра (замены) или отмены настоящего стандарта*

соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя "Национальные стандарты". Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования - на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (www.gost.ru)

Введение

Информационное моделирование объектов строительства представляет собой концепцию для описания и представления информации, необходимой для проектирования, строительства и управления построенными объектами. Эта концепция объединяет различные наборы используемых в строительстве сведений в единую информационную среду, уменьшая или даже исключая необходимость использования многих видов бумажной документации, традиционно используемых в настоящее время.

Справочник по обмену информацией (IDM) значительно способствует наиболее эффективному применению информационной модели здания (BIM). Быстрый доступ к необходимой информации хорошего качества значительно улучшает строительный процесс. Для этого требуется единое понимание строительных процессов и информации, необходимой для их выполнения.

В настоящем стандарте основное внимание уделяется аспектам строительного процесса, относящимся к задачам управления участниками процесса и координации их действий. Координация, в свою очередь, зависит от коммуникации, которая должна быть хорошо структурированной, понятной, исчерпывающей и оперативной. Благодаря акценту на координацию и взаимодействие настоящий стандарт естественным образом дополняет такие стандарты моделирования зданий, как ИСО 10303-239 и ИСО 16739-1:2018.

В настоящем стандарте излагаются методология и формат описания действий по координации акторов в строительном проекте. В нем описывается, как выявлять и определять координационные процессы и необходимую для их выполнения информацию. Получаемая в результате структура взаимодействия позволяет стандартизировать это взаимодействие в строительном проекте на национальном, локальном и проектном уровнях. Здесь также предлагается формат для поддержки решений, предоставляемых поставщиками ИКТ-продуктов. Поддержка различными ИКТ-решениями настоящего стандарта подразумевает объединение различных систем управления процессами. Таким образом, настоящий стандарт представляет собой основу для надежного обмена информацией и ее совместного использования.

Разработка настоящего стандарта вызвана потребностью участников процесса в надежности при обмене информацией. Она основывается главным образом на стандарте Нидерландов VISI, разработанном в 2003 году.

1 Область применения

В настоящем стандарте представлены методология и формат для описания действий по координации между акторами строительного проекта на всех этапах его жизненного цикла.

В настоящем стандарте приведены:

- методология, описывающая структуру взаимодействия,

- соответствующий способ сопоставления обязанностей и взаимодействий, создающих контекст процесса для потока информации,

- формат, в котором должна описываться структура взаимодействия.

Настоящий стандарт предназначен содействовать интероперабельности между используемыми в процессе строительства программными средствами, переводу взаимодействия между акторами процесса строительства на платформу цифровых информационных технологий и созданию основы для точного, надежного, повторяемого и высококачественного обмена информацией.

2 Нормативные ссылки

В настоящем стандарте использована нормативная ссылка на следующий стандарт:

ISO 29481-1, Building information models - Information delivery manual - Part 1: Methodology and format (Информационное моделирование в строительстве. Справочник по обмену информацией. Часть 1. Методология и формат)

3 Термины и определения

В настоящем стандарте применены следующие термины с соответствующими определениями:

3.1 **справочник по обмену информацией** (Information Delivery Manual, IDM): Документация, описывающая бизнес-процесс и содержащая подробное описание информации, которую на определенном этапе проекта должен предоставить пользователь, выполняющий определенную роль.

3.2 **структура взаимодействия** (interaction framework): Формальное описание элементов взаимодействия, включая определение ролей, транзакций, сообщений в транзакциях и элементов данных в сообщениях.

3.3 **схема структуры взаимодействия** (interaction framework schema): Формальное описание правил, которым должна подчиняться структура взаимодействия.

3.4 **схема взаимодействия** (interaction schema): Формальное описание правил, которым должны соответствовать отправленные и полученные сообщения.

3.5 **промотор** (promotor): Алгоритм, генерирующий схему взаимодействия из структуры взаимодействия, схемы структуры взаимодействия и файла шаблонов в качестве входных данных.

3.6 **файл шаблонов** (templates file): Файл, содержащий несколько независимых от структуры взаимодействия шаблонов для формирования схемы взаимодействия.

3.7 **VISI:** Аббревиатура, обозначающая стандарт Нидерландов по взаимодействию между участниками в строительных проектах.

Примечание - VISI означает "Voorwaarden scheppen voor Invoeren Standaardisatie ICT in de Infrastructuur-sector", что переводится как "Создание условий для стандартизации ИКТ-технологий в строительной отрасли".

4 Основные принципы

4.1 Общие положения

Настоящий раздел выделяет и объясняет основные понятия, на которых основан настоящий стандарт.

4.2 BIM и IDM

Информационное моделирование зданий объединяет различные используемые в строительстве наборы информации в единую информационную среду. Для этого необходимо единое понимание строительных процессов и информации, необходимой для их выполнения и получаемой в качестве результата.

В настоящем стандарте излагается метод разработки справочника по обмену информацией. Приведенная в ИСО 29481-1:2016 методология IDM должна использоваться для всех упоминаний о разработке и использовании IDM.

4.3 Компоненты IDM

Методология и компоненты IDM описаны в ИСО 29481-1:2016. В указанном стандарте схематически показано, что представляют собой различные компоненты IDM и как они связаны между собой.

IDM можно рассматривать под двумя углами: в ракурсе пользовательских требований и в контексте технических решений. Для каждого подхода выделяется ряд зон, характеризующих различные компоненты IDM (см. рисунок 1).

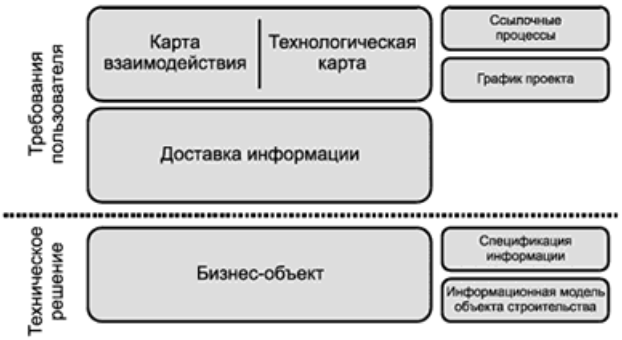

Рисунок 1 - Зоны IDM

С точки зрения пользовательских требований к этим зонам относятся:

- карты взаимодействия, описывающие роли и взаимодействия между ними;

- технологические карты, описывающие общий процесс, в котором происходит обмен информацией;

- доставка информации, описывающая потребности в обмене информацией;

- ссылочные процессы (хранимые описания операций обмена информацией);

- график проекта (реализации процессов в контексте проекта).

С точки зрения технических решений выделяют следующие зоны:

- бизнес-объекты, включающие перечень требований к обмену информацией;

- спецификация информации, описывающая схему, лежащую в основе обмена информацией;

- информационная модель здания.

Настоящий стандарт регламентирует карты взаимодействия и основывается на общих принципах деловой коммуникации.

4.4 Основные принципы деловой коммуникации

После запроса клиента или заказчика предоставить продукт или оказать услугу возникает цепочка действий, совместный результат которых заключается в предоставлении продукта или услуги. Такая цепочка действий называется бизнес-процессом. Речь идет, в частности, о первичном бизнес-процессе, поскольку он инициируется извне.

Одной из составляющих бизнес-процесса является общение или коммуникация между вовлеченными сторонами. Настоящий стандарт посвящен в основном коммуникации, связанной с предоставлением результата (исполнительная коммуникация). Инициация и выполнение запроса осуществляются посредством коммуникативных действий. В коммуникативном действии всегда задействованы две стороны: лицо, совершившее действие, и лицо, которому это действие было направлено. Обработка запроса происходит по определенному шаблону, называемому транзакцией.

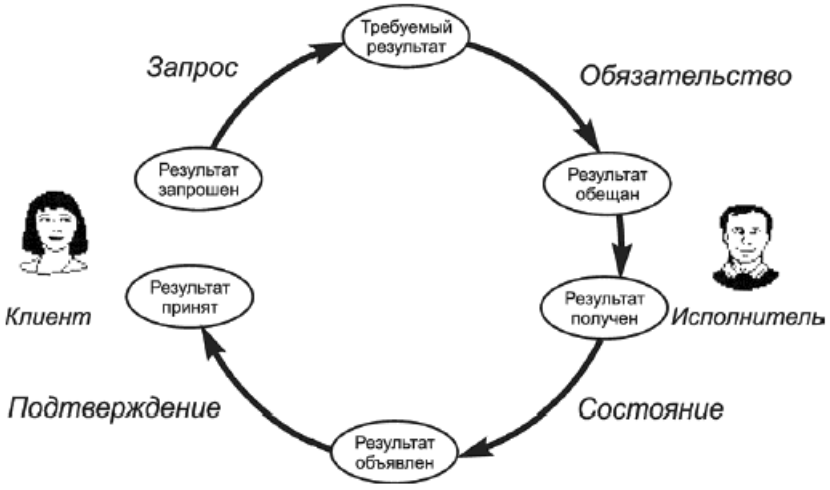

Рисунок 2 - Шаблон транзакции (Dietz J.L.G., 2006 [1])

На рисунке 2 представлена простейшая форма шаблона транзакции. Шаблон показывает, что достижение какого-либо нового результата (возьмем в качестве "желаемого результата", например, предоставление некоего документа) начинается с запроса этого результата лицом, действующим в роли заказчика, у лица, действующего в роли исполнителя. Это действие переводит процесс в состояние "результат запрошен". Далее исполнитель отвечает на запрос, пообещав предоставить желаемый результат, тем самым переводя процесс в состояние "результат обещан". Таким образом, у исполнителя появляется задача: он должен выполнить обещание, фактически подготовив документ и приняв решение предоставить его заказчику. При передаче документа заказчику исполнитель сообщает, что его обещание выполнено. Заказчик отвечает путем приемки полученного результата. Это действие завершает транзакцию.

В бизнес-процессе зачастую участвует множество акторов. Их поведение зависит от их роли в конкретном процессе. Роли/акторы взаимодействуют с другими ролями/акторами путем осуществления транзакций. Удобным представлением взаимодействия между ролями/акторами является карта взаимодействия.

4.5 Карта взаимодействия

Карта взаимодействия определяет соответствующие типы ролей и транзакций для определенного процесса. IDM выделяет роль, делающую запрос (инициатор), и роль, отвечающую на этот запрос (исполнитель). В каждой транзакции может быть только одна инициирующая роль и одна исполняющая. На рисунке 3 показаны компоненты карты взаимодействия.

Примечание - Система обозначений на карте взаимодействия основана на строительной модели, описанной в публикации профессора Яна Л.Г.Дитца. Она отличается от системы обозначений BPMN и используется для создания максимально простых карт. Также она включает концепцию "транзакции", отсутствующую в BPMN.

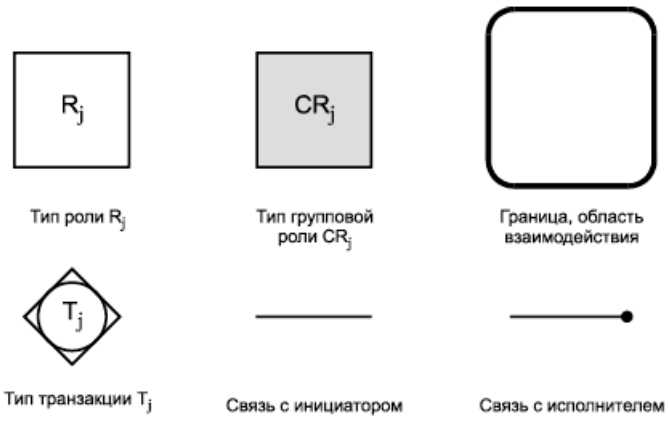

Рисунок 3 - Компоненты карты взаимодействия

Преимущество карты взаимодействия заключается в том, что она фокусируется на интерфейсах между ролями, скрывая при этом сложность конкретного процесса в области ролей и подробности взаимодействия между этими ролями. Использование абстрактных ролей позволяет применять карту взаимодействия в самых разных случаях. Карта взаимодействия - это ценный инструмент для анализа и определения основных элементов бизнес-процесса. На рисунке 4 показан упрощенный пример карты взаимодействия конструкторского бюро.

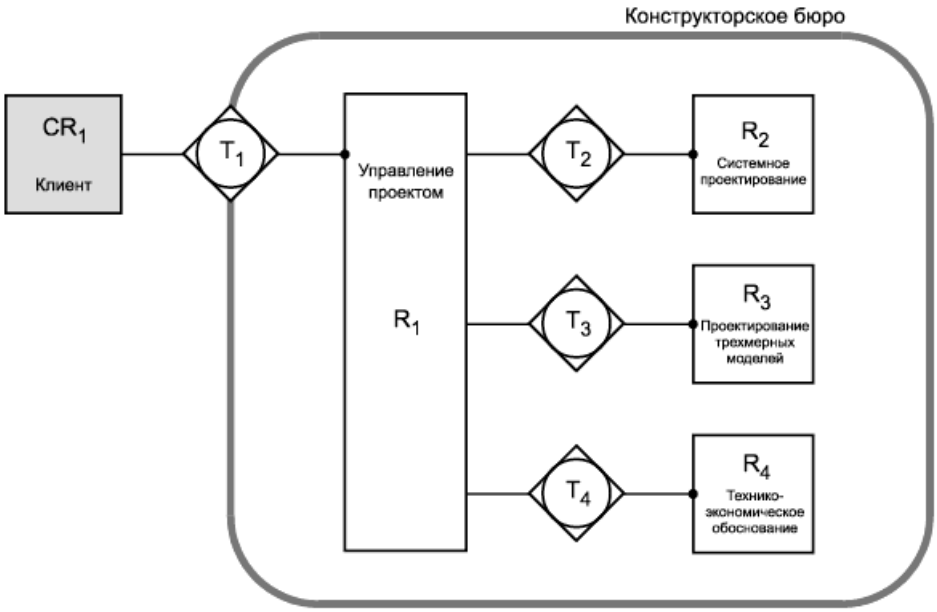

Рисунок 4 - Пример карты взаимодействия

В карту взаимодействия должны быть включены все транзакции, необходимые для обработки требуемых вкладов соответствующих ролей в процесс BIM. У всех ролей и транзакций в карте взаимодействия должны быть уникальный идентификатор и имя. Нумерация при этом произвольная. Название роли определяется основным действием, выполняемым ролью, что делает акцент на вкладе определенной роли в BIM. Составная роль - это роль, которая может состоять из нескольких ролей, состав которых неизвестен или не имеет особого значения.

Обобщенная информация о взаимодействиях может быть представлена в таблице транзакций.

Таблица 1 - Упрощенная таблица транзакций конструкторского бюро

Результат транзакции	Тип транзакции
Разработан проект	T1, разработка проекта
Разработана спецификация системы	T2, разработка спецификации системы
Разработана трехмерная модель	T3, разработка трехмерной модели
Разработана смета	T4, разработка сметы

4.6 Сообщения в транзакции

Транзакция содержит набор сообщений, обмен которыми происходит для определенной цели. Транзакция также устанавливает участвующие роли, жизненный цикл и последовательность, в которой должны доставляться сообщения (при необходимости).

В качестве примера транзакции можно привести обработку запроса трехмерной модели. На рисунке 5 показаны сообщения в транзакции в форме диаграммы последовательности в обозначении UML. Транзакция может быть инициирована только руководителем проекта R1 с помощью сообщения "Запрос трехмерной модели". Инженер - разработчик трехмерных моделей (роль R3) может направить в ответ сообщение "Задание выполнено, запрашивается утверждение выполнения". После отправки сообщения "Выполнение задания утверждено" (или "Выполнение задания не утверждено") транзакция будет завершена.

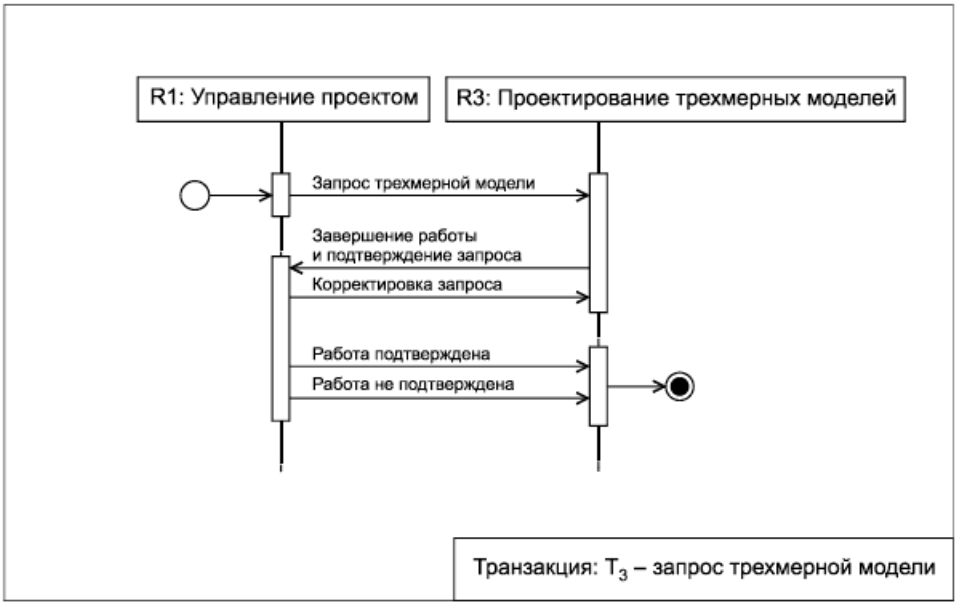

Рисунок 5 - Пример сообщений в транзакции

Сообщение представляет собой заполненную информационную модель, содержащую данные. К сообщениям могут прилагаться вложения. Требование к обмену может быть передано как вложение исполняющей роли, а результат (вклад в BIM) направляется инициирующей роли. С помощью транзакций передача информации осуществляется в контексте процесса.

4.7 Структура взаимодействия

Для выдачи указаний процессу и передачи информации элементы взаимодействия необходимо описать упорядоченным образом. Такое упорядоченное описание носит название структуры взаимодействия. Структура взаимодействия включает:

- определение соответствующих ролей,

- транзакции,

- сообщения в транзакции,

- порядок сообщений в транзакции,

- элементы данных в сообщениях.

Структура взаимодействия может быть подготовлена для определенной области применения и использоваться в качестве стандарта на межнациональном (национальном) уровне, уровне организации или уровне проекта. Например, в Нидерландах на национальном уровне разрабатывается структура взаимодействия для выполнения всех договорных процедур при реализации строительного проекта. Настоящий стандарт используется в качестве шаблона организациями и проектами и адаптируется к конкретным потребностям.

Пример - Структура взаимодействия может включать атрибут CostEstimation в качестве экземпляра SimpleElementType, который будет использоваться в качестве обязательного элемента для определенного сообщения. Она также может включать ограничение для формата атрибута CostEstimation (например, только евро с двумя десятичными знаками).

4.8 Поддержка программных решений

4.8.1 Обзор

Следующим шагом является поддержка структуры взаимодействия с программными решениями в следующих целях:

- поддержка редактирования структуры взаимодействия,

- гарантия полноты и обоснованности структуры взаимодействия,

- поддержка портативности структуры взаимодействия,

- поддержка работы информационных систем,

- поддержка коммуникационной интероперабельности.

Поддержка программных решений осуществляется на двух уровнях. Первый уровень относится к структуре взаимодействия. Второй уровень относится к фактической коммуникации, которая осуществляется на основе структуры взаимодействия. Настоящий стандарт применим к обоим уровням.

На рисунке 6 показано, как осуществляется поддержка программных решений. В следующих разделах приводится пояснение.

4.8.2 Поддержка структуры взаимодействия

Чтобы обеспечить портативность структуры взаимодействия, необходимо четко обозначить, каким правилам она должна соответствовать. Эти правила должны быть включены в схему структуры взаимодействия, которая записывается в XSD-файл схемы. Структура взаимодействия включает в себя экземпляры классов, определенных в схеме, и записывается в XML-файл.

Пример - Схема структуры взаимодействия определяет, какие определения атрибутов (SimpleElementType) и ограничений на атрибуты (UserdefinedType) вы можете включить в структуру взаимодействия.

В разделе 5 приведено описание схемы структуры взаимодействия и доступных классов.

Каждая структура взаимодействия должна соответствовать схеме структуры взаимодействия.

Редактор структуры взаимодействия должен использовать схему структуры взаимодействия для валидации созданных структур.

4.8.3 Промотор

Как только действительная структура взаимодействия станет доступной, ее можно будет интерпретировать с помощью подходящей информационной системы. Затем эта система сможет поддерживать коммуникации в соответствии с параметрами, заданными в структуре взаимодействия. И наконец, желательно, чтобы имелась возможность проводить валидацию принимаемых и отправляемых сообщений, что осуществляется с помощью схемы взаимодействия.

Схема взаимодействия генерируется с помощью общего алгоритма, называемого "промотор". Промотор "продвигает" экземпляры XML в классы XSD. В качестве входных данных используются:

- структура взаимодействия (XML),

- схема структуры взаимодействия (XSD),

- файл шаблонов (XSD), содержащий несколько шаблонов, не описанных в структуре взаимодействия, но действительных для каждой схемы взаимодействия, например шаблон заголовка сообщения.

На выходе получаем схему взаимодействия в виде XSD-файла.

Пример - Промотор получает информацию из структуры взаимодействия, чтобы включить атрибут CostEstimation, который будет использоваться как обязательный элемент для определенного сообщения, и создает схему взаимодействия, которая определяет сообщение с атрибутом CostEstimation.

В приложении B описываются шаблоны, содержащиеся в XSD-файле.

В приложении D описываются принципы работы промотора.

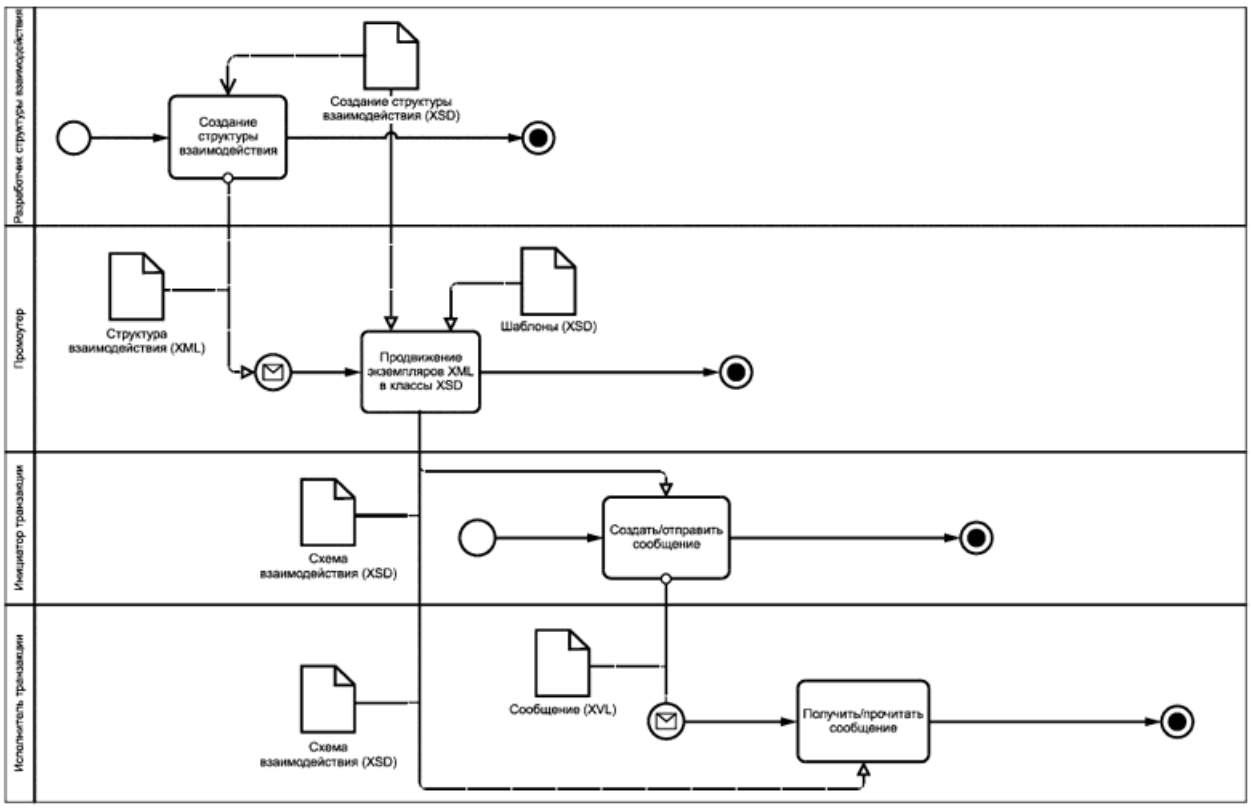

Рисунок 6 - Схема поддержки программных решений

4.8.4 Поддержка обмена информацией

Каждая информационная система, участвующая в обмене информацией согласно правилам, определенным в структуре взаимодействия, должна работать на основе соответствующих структуры взаимодействия и схемы взаимодействия. Каждое отправленное или принятое сообщение должно быть действительным согласно схеме взаимодействия.

4.8.5 Техническая реализация обмена информацией

Чтобы обеспечить техническую сторону обмена сообщениями с вложениями между информационными системами, должны быть предусмотрены указания по его реализации. Они должны охватывать следующее:

- протокол обмена информацией,

- архитектуру/сервер обмена информацией,

- шифрование,

- вызов SOAP-функции.

Указания по реализации не входят в область рассмотрения настоящего стандарта.

5 Формат структуры взаимодействия

5.1 Введение

Как указано в разделе 4, для поддержки программных решений каждая структура взаимодействия должна соответствовать схеме структуры взаимодействия. Этот раздел определяет формат структуры взаимодействия посредством описания схемы структуры взаимодействия.

В подразделе 5.2 представлен обзор классов информации, которые могут встречаться в структуре взаимодействия и определены в схеме структуры взаимодействия. Поскольку структура взаимодействия определена в XML, используется термин "тип", а не "класс". В приложении А приведено полное описание схемы структуры взаимодействия. В приложении В приведен пример экземпляра структуры взаимодействия.

5.2 Типы информации в схеме структуры взаимодействия

5.2.1 Введение

Схема заполняется рядом классов или типов. В этом разделе приводится краткое описание доступных типов в схеме структуры взаимодействия. В приложении А содержится полное описание схемы структуры взаимодействия в XML. Структура взаимодействия создается из экземпляров этих типов и имеет заголовок, который указывает на схему с определенными доступными типами.

5.2.2 AppendixType

AppendixType - это определение, которое определяет структуру элементов, относящихся к метаданным. Экземпляр AppendixType используется для определения определенных типов файлов или документов, которые могут быть частью отправляемых/получаемых сообщений. Структура элементов, связанных с экземпляром AppendixType, представляет определенные метаданные, которые необходимы для определенного типа файла или документа.

5.2.3 ComplexElementType

ComplexElementType содержит набор SimpleElementTypes. Каждый тип SimpleElementType встречается ровно столько раз, сколько встречается элемент, к которому он относится.

5.2.4 ElementCondition

Экземпляр ElementCondition описывает поведение значений элементов в последовательности сообщений. Например, когда экземпляр типа ElementCondition создается со значением FIXED, это указывает на то, что элементы в последовательности сообщений должны копироваться в случаях, когда доступен один и тот же элемент и его значение не может быть изменено. ElementCondition может ссылаться на различные уровни в структуре. Он может быть напрямую связан с SimpleElement, но также можно связать ElementCondition с ComplexElement или MessageInTransactionType. В этом случае ElementCondition является действительным для всех элементов, которые являются частью структуры/набора элементов связанных типов.

5.2.5 GroupType

Тип GroupType позволяет создавать несколько экземпляров группы с собственным специфическим содержимым. GroupType можно использовать для категоризации сообщений в транзакции или документов, связанных с сообщениями в транзакции.

5.2.6 MessageType

Тип MessageType используется для определения содержимого сообщения. Элементы, которые являются частью сообщения, в свою очередь группируются в один или несколько экземпляров ComplexElementType.

5.2.7 MessageInTransactionType

MessageInTransactionType (MITT) представляет собой определение, используемое для связывания экземпляров типа MessageType с экземплярами типа TransactionType. Проще говоря, один и тот же тип сообщения может встречаться в данном типе транзакции чаще, чем один раз, и наоборот. Можно связать несколько экземпляров MessageType с одним экземпляром TransactionType и один экземпляр MessageType с несколькими экземплярами TransactionType. Кроме того, MITT дает возможность изменить направление действия от исполнителя к инициатору. Также он предусматривает возможность отслеживания того, блокируется ли поток сообщений открытыми вторичными транзакциями или нет.

5.2.8 OrganisationType

Определение конкретной группы организаций. В общем случае в структуре доступен как минимум один экземпляр с конкретной причиной для определения структуры элементов организации.

5.2.9 PersonType

Определение типа лица. В общем случае в структуре доступен как минимум один экземпляр с конкретной причиной для определения структуры элементов, определяющих лицо. Тип PersonType может использоваться для классификации групп лиц, связанных с какой-либо ролью.

5.2.10 ProjectType

Определение типа проекта. Говоря в общем, в структуре доступен как минимум один экземпляр с конкретной причиной для определения структуры элементов, определяющих проект.

5.2.11 RoleType

Определение роли. Экземпляры типа RoleType необходимы для создания TransactionType в структуре.

5.2.12 SimpleElementType

SimpleElementType - это определение, описывающее свойства, которые могут встречаться в структурах объектов. Связь с объектом всегда устанавливается через ComplexElementType.

5.2.13 TransactionPhaseType

Определение, которое может использоваться для определения экземпляра, описывающего этап транзакции. Например, экземпляр TransactionPhaseType может соответствовать этапам "запрошен результат" или "в ожидании".

5.2.14 TransactionType

Определение транзакции. В экземпляре транзакции определяются роли инициатора и исполнителя.

5.2.15 userDefinedType

Тип UserDefinedType используется для определения типов данных (например, строки) и ограничений XSD. Например, с помощью экземпляра UserDefinedType можно определить минимальную длину строки.

На рисунке 7 показано графическое отображение модели схемы структуры взаимодействия, включая все ссылки.

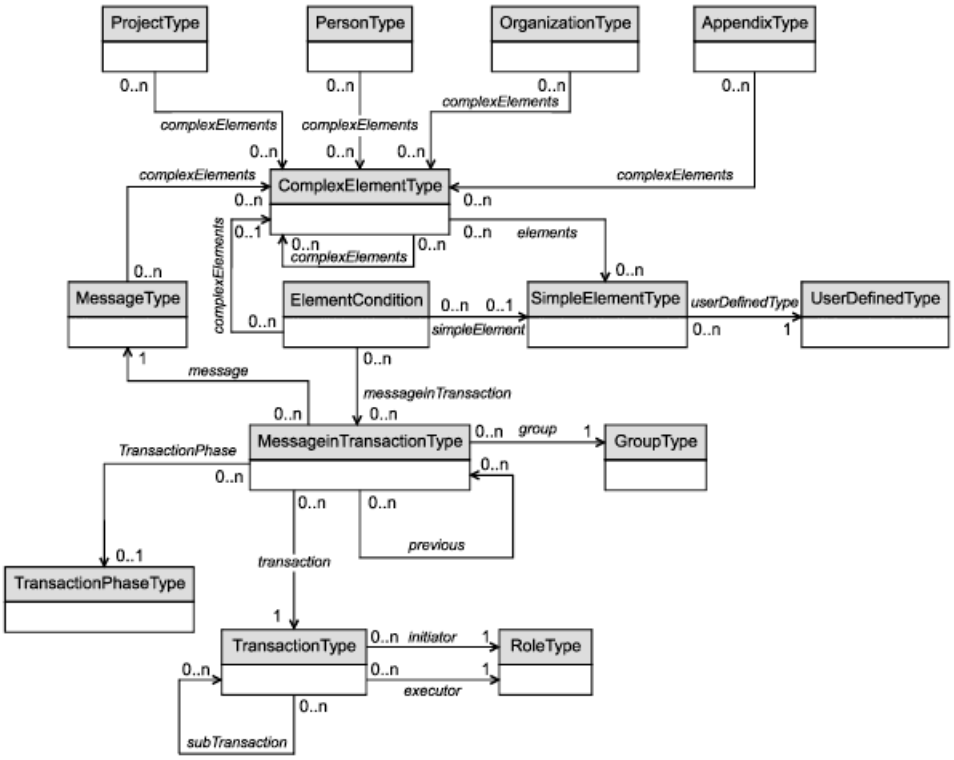

Рисунок 7 - Типы и ссылки схемы структуры взаимодействия

Приложение А
(обязательное)

Определение схемы структуры взаимодействия

А.1 Введение

Структура взаимодействия должна соответствовать правилам, которые включены в схему структуры взаимодействия. В настоящем приложении определение схемы структуры взаимодействия приводится в формате EXPRESS и в формате XSD. Кроме того, приведены описания типов элементов, атрибутов, элементов и ссылок.

Для каждого объекта в схеме взаимодействия требуются дата начала и дата окончания. Это позволяет включать временные ограничения в срок действия конкретного объекта.

А.2 Определение схемы структуры взаимодействия (формат EXPRESS) 1)

_____

1) В настоящем стандарте приводится полное описание схемы структуры взаимодействия, представленное в ИСО 29481-2:2012.

```
SCHEMA ISO 29481_Part_2A;

ENTITY ProjectType; - Определение конкретной группы проектов. Обычно в структуре взаимодействия будет присутствовать только один экземпляр, определяющий структуру элементов, которые будут предоставляться в каждом экземпляре проекта.

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; - Определение конкретной группы лиц. Обычно в структуре взаимодействия будет присутствовать только один экземпляр, определяющий структуру элементов, которые будут предоставляться в каждом экземпляре лица.

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; - Определение конкретной группы организаций. Обычно в структуре взаимодействия присутствует только один экземпляр, определяющий структуру элементов, которые должны предоставляться в каждом экземпляре организации.

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; - AppendixType содержит определение дополнения. Какие элементы данных должны записываться с помощью дополнения, можно указать в разделе составного элемента.

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; - ComplexElementType содержит набор SimpleElementTypes. Каждый заявленный тип SimpleElementType входит ровно столько раз, сколько он упоминается.

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; - Определение типа сообщения (MessageType), которое указывает структуру сообщения и какой набор типов SimpleElementType (через ComplexElementType) оно может сопровождать.

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; - Условие SimpleElementType в том виде, как оно используется в конкретном MessageType.

description: STRING;

condition: STRING;

helpInfo: OPTIONAL STRING;

complexElement: OPTIONAL ComplexElementType;

simpleElement: OPTIONAL SimpleElementType;

messageInTransaction: OPTIONAL MessageInTransactionType;

END_ENTITY;

ENTITY SimpleElementType; - Определение типа простого элемента (SimpleElementType). Этот тип элемента указывает свойство, которое может встречаться в различных структурах объекта, например в MessageType (см. также AppendixType, ProjectType, PersonType и OrganisationType). SimpleElementType всегда является вложенным в 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; - Спецификация типа данных (для использования в SimpleElementType). Этот тип данных формирует заполняемые области в конечном сообщении; например, нидерландский почтовый индекс всегда начинается с четырех цифр, за которыми следуют две буквы.

description: STRING;

state: STRING;

dateLaMu: DATETIME;

userLaMu: STRING;

baseType: STRING;

xsdRestriction: OPTIONAL STRING;

language: OPTIONAL STRING;

helpInfo: OPTIONAL STRING;

END_ENTITY;

ENTITY MessageInTransactionType; - Образец типа MessageType в типе TransactionType, относящийся к конкретному типу группы (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; - Определение фазы, относящейся к транзакции.

Примеры: "назначение принято" или "часть результата получена".

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; - Определение типа транзакции. Тип транзакции может ссылаться на другие типы транзакции. Транзакцию будет инициировать лицо, принадлежащее к организации, выполняющей определенную роль. На этом уровне должен быть заявлен тип роли инициатора (введенный в действие активированной схемой). Это же верно для исполнителя.

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; - Определение конкретного типа роли.

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;

esponsibilityFeedback: OPTIONAL STRING;

END_ENTITY;

ENTITY GroupType; - Определение группы для сохранения дополнений, отправленных с сообщением в транзакции.

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;

А.3 Определение схемы структуры взаимодействия (формат XSD)

```
<?xml version="1.0" encoding = "UTF-8"? >
<xsd:schema targetNamespace="http://www. ISO 29481_Part_2A.com/senemas"
xmlns:iso29481p2a = "http://www.ISO 29481_Part_2A.com/schemas"
xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
elementFormDefault = "qualified" >
<!- объявление корневого элемента (для SCHEMA определений) →
<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>
```

<!- объявление элемента (для ENTITY определений) →
<xsd:element name="AppendixType"
type = "iso29481p2a:AppendixTypeType" >
<xsd:annotation>
<xsd:documentation>Тип AppendixType содержит определение приложения. Какие элементы данных должны быть записаны в приложении, можно указать в разделе сложный элемент.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ComplexElementType"
type = "iso29481p2a:ComplexElementTypeType" >
<xsd:annotation>
<xsd:documentation>Сложный тип элемента ComplexElementType содержит набор простых типов элементов SimpleElementTypes. Каждый, из описанных SimpleElementType встречается ровно столько раз, сколько указано.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ElementCondition"
type = "iso29481p2a:ElementConditionType" >
<xsd:annotation>
<xsd: documentation>Условие SimpleElementType, используемое в определенном MessageType.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="GroupType" type = "iso29481p2a:GroupTypeType" >
<xsd:annotation>
<xsd:documentation>Определение группы для хранения приложений, отправленных с сообщением в транзакции.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd: element name="MessageInTransactionType"
type = "iso29481p2a:MessageInTransactionTypeType" >
<xsd:annotation>
<xsd:documentation>Возникновение MessageType в TransactionType, связанного с определенным типом группы (GroupType).</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="MessageType" type = "iso29481p2a:MessageTypeType" >
<xsd:annotation>
<xsd:documentation>Определение типа сообщения (MessageType), указывающее структуру сообщения и какой набор SimpleElementType (через ComplexElementType) оно может сопровождать.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd: element name="OrganisationType"
type = "iso29481p2a:OrganisationTypeType" >
<xsd:annotation>
<xsd:documentation>Определение конкретной группы организаций. Как правило, только один экземпляр будет присутствовать в структуре взаимодействия, определяющей структуру элементов, которые должен представлять каждый экземпляр организации.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd: element name="PersonType" type = "iso29481p2a:PersonTypeType" >
<xsd:annotation>
<xsd:documentation>Определение конкретной группы людей. Как правило, только один экземпляр будет присутствовать в структуре взаимодействия, определяющей структуру элементов, которые должен представлять каждый экземпляр человека.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ProjectType" type = "iso29481p2a:ProjectTypeType" >
<xsd:annotation>
<xsd:documentation>Определение конкретной группы проектов. Как правило, только один экземпляр будет присутствовать в структуре взаимодействия, определяющей структуру элементов, которые должен представлять каждый экземпляр проекта.</xsd:documentation>

```
</xsd:annotation>
</xsd:element>
<xsd: element name="RoleType" type = "iso29481p2a:RoleTypeType" >
<xsd:annotation>
<xsd:documentation>Определение конкретного типа роли.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd: element name="SimpleElementType"
type = "iso29481p2a:SimpleElementTypeType" >
<xsd:annotation>
<xsd:documentation>Определение простого типа элемента (SimpleElementType). Этот тип элемента определяет свойство, которое может встречаться в различных структурах объектов, например, в MessageType (см. также AppendixType, ProjectType, PersonType и OrganisationType).
SimpleElementType всегда встроен в
ComplexElementType.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TransactionPhaseType"
type = "iso29481p2a:TransactionPhaseTypeType" >
<xsd:annotation>
<xsd:documentation>Определение фазы, связанной с транзакцией. Примерами являются "принятое назначение" или "полученная часть результата".</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TransactionType"
type = "iso29481p2a:TransactionTypeType" >
<xsd:annotation>
<xsd:documentation>Определение типа транзакции. Тип транзакции может вновь ссылаться на другие типы транзакций. Инициатором сделки является лицо, принадлежащее к организации, выполняющей определенную роль. На этом уровне должен быть указан тип роли инициатора (продвинутая схема будет применять это). То же самое относится и к исполнителю.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="UserDefinedType"
type = "iso29481p2a:UserDefinedTypeType" >
<xsd:annotation>
<xsd:documentation>Спецификация типа данных (для использования в SimpleElementType).</xsd:documentation>
</xsd:annotation>
</xsd:element>
<!- объявления ссылок на элементы (для ENTITY определений) →
<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:elementname="TransactionPhaseTypeRef"
type = "iso29481p2a:Transact ionPhaseTypeTypeRef"/ >
<xsd:element name="TransactionTypeRef"
type = "iso29481p2a:TransactionTypeTypeRef"/ >
<xsd:element name="UserDefinedTypeRef"
type = "iso29481p2a:UserDefinedTypeTypeRef"/ >
<!- объявление ссылок на сложные элементы (для ENTITY определений) →
<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:element Type" >
<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"/ >
<xsd:element name="firstMessage"
type = "xsd:boolean"
minOccurs = "0"/ >
<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:element ref="iso29481p2a:AppendixTypeRef"/ >
</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="SimpleElementTypeType" >
<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="TransactionPhaseTypeType" >
<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:complexType>
<xsd:complexType name="TransactionTypeType" >
<xsd:complexContent>
<xsd: extension base="iso29481p2a:element Type" >
<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:element Type " >
<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>
<!- объявление ссылок на сложные элементы (для ENTITY определений) →
<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>
<!- группа объявлений (для SELECT определения типов данных) →
<!- объявления типов перечислений (для ENUMERATION определения типов данных) →
<!- объявления простого типа (для TYPE заданных определений типов данных) →
<!- стандартные декларации, elementType, simpleType, logical (для каждой трансляции) →
<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 Типы элемента

A.4.1 AppendixType

Атрибуты: id

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

Ссылки: complexElements

AppendixType содержит определение дополнения. Какие элементы данных должны записываться с помощью дополнения, можно указать в разделе составного элемента.

Пример -

```
<AppendixType id="StandardAppendixType" >
<description>Стандартный тип приложения</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>
```

Примечание - Элемент **appendixTypes**. Если структура взаимодействия указывает много типов дополнений, пользователям может быть трудно выбрать подходящий тип дополнения. Эта ссылка позволяет выбирать из набора всех типов дополнений типы, действительные для конкретных транзакции или сообщения.

A.4.2 ComplexElementType

Атрибуты: id

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

Ссылки: elements

ComplexElementType содержит набор SimpleElementTypes. Каждый заявленный тип SimpleElementType входит ровно столько раз, сколько он упоминается.

Пример -

```
<ComplexElementType id="MenuItem" >
<description>Пункт меню</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

Атрибуты: id

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

Ссылки: message, element

Условие SimpleElementType в том виде, как оно используется в конкретном MessageType.

Пример -

```
<ElementCondition id="Price restriction" >
<discription>Минимальная цена пункта меню</description>
<requiredNotify>0</requiredNotify>
<minValue>5.00</minValue>
<element>
<SimpleElementType>Цена</SimpleElementType>
</element>
<message>
<MessageTypeRef idref="ProvideWithMenuMessage"/ >
</message>
</ElementCondition>
```

A.4.4 GroupType

Атрибуты: id

Элементы: description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpInfo. Определение группы для сохранения дополнений, отправленных вместе с сообщением в транзакции.

Пример -

```
<GroupType id="StandardGroupType" >
<description>Стандартная группа</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

Атрибуты: id

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

Ссылки: complexElements, appendixTypes

Определение типа сообщения (MessageType), которое указывает структуру сообщения и какой набор типа SimpleElementType (через ComplexElementType) оно может сопровождать.

Пример -

<MessageType id="ProvideWithMenuMessage" >
<description>Сообщение, содержащее меню.</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>

Примечание - Элемент **firstMessage**. В общем случае транзакция будет запускаться лицом, выполняющим роль инициатора, с помощью типа сообщения, независимого от типа любого предыдущего сообщения. Однако этот принцип нельзя применять к транзакции, которая выполняется как подтранзакция в другой транзакции. Этот элемент явно заявляет, можно ли использовать тип сообщения для запуска новой транзакции.

A.4.6 MessageInTransactionType

Атрибуты: id

Элементы: requiredNotify, dateLaMu, userLaMu, received, send, state, initiatorToExecutor, openSecondaryTransactionsAllowed, firstMessage

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

Образец типа MessageType в типе TransactionType, относящийся к конкретному типу группы (GroupType).

Пример -

<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

Атрибуты: id

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

Ссылки: complexElements

Определение конкретной группы организаций. Обычно в структуре взаимодействия присутствует только один экземпляр, определяющий структуру элементов, которые должны предоставляться в каждом экземпляре организации.

Пример -

```
<OrganisationType id="StandardOrganisationType" >
<description>Стандартный тип организации</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

Атрибуты: id

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

Ссылки: complexElements

Определение конкретной группы лиц. Обычно в структуре взаимодействия будет присутствовать только один экземпляр, определяющий структуру элементов, которые будут предоставляться в каждом экземпляре лица.

Пример -

```
<PersonType id="StandardPersonType" >
<description>Стандартный тип физического лица</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

Атрибуты: id

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

Ссылки: complexElements

Определение конкретной группы проектов. Обычно в структуре взаимодействия будет присутствовать только один экземпляр, определяющий структуру элементов, которые будут предоставляться в каждом экземпляре проекта.

Пример -

<ProjectType id="StandardProjectType" >
<namespace>http://www.ISO 29481_Part_2A.com/testproject</namespace>
<description>Стандартный тип проекта</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

Атрибуты: id

Элементы: description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpInfo, code, responsibilityFeedback, responsibilityScope, responsibilitySupportTask, responsibilityTask

Определение конкретного типа роли.

Пример -

<RoleType id="waiter" >
<description>Ответственный за прием и размещение заказов</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

Атрибуты: id

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

Ссылки: composedOf, userDefinedType

Определение типа простого элемента (SimpleElementType). Этот тип элемента указывает свойство, которое может встречаться в различных структурах объектов, например в MessageType (см. также AppendixType, ProjectType, PersonType и OrganisationType). SimpleElementType всегда является вложенным в ComplexElementType.

Пример -

```
<SimpleElementType id="Name" >
<description>Название пункта меню</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

Атрибуты: id

Элементы: description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helplnfo, code Thedefinitionofaphaserelated toatransaction. Примеры: "назначение принято" или "часть результата получена".

Пример -

```
<TransactionPhaseType id="WaitingForMenu" >
<description>Меню запрошено, но еще не доставлено</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

Атрибуты: id

Элементы: description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helplnfo, code, result

Ссылки: initiator, executor, subTransactions

Определение типа транзакции. Тип транзакции может в свою очередь ссылаться на другие типы транзакции. Транзакцию будет инициировать лицо, принадлежащее к организации, выполняющей определенную роль. На этом уровне должен быть заявлен тип роли инициатора (продвигаемая схема будет обеспечивать это). Это же верно для исполнителя.

Пример -

```
<TransactionType id="MenuObtainingTransaction" >
<description>Транзакции для получения нужого меню</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="Заказчик"/ >
</initiator>
<executor>
<RoleTypeRef idref="Сотрудник"/ >
```

```
</executor>
</TransactionType>
```

A.4.14 UserDefinedType

Атрибуты: #id

Элементы: description, state, dateLaMu, userLaMu, baseType, xsdRestriction, language, helpInfo

Спецификация типа данных (для использования в SimpleElementType). Этот тип данных инкапсулирует заполняемые области в конечном сообщении.

Пример -

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

A.5 Атрибуты

A.5.1 id

Уникальное "краткое" имя образца. После продвижения это имя будет именем объекта.

Пример -

Объект структуры

```
<OrganisationType id="TNO Building and Construction" >
<description>Атрибуты организации</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 Элементы

A.6.1 baseType

Содержит базовый тип типа SimpleElementType.

Пример -

```
<SimpleElementType id="Height" >
…
<userDefinedType>
<UserDefinedType id="..." >
…
```

```
<baseType>INTEGER</baseType>
…
</UserDefinedType>
</UserDefinedType>
</SimpleElementType>
```

Здесь *Height* в типе SimpleElementType всегда содержит целое число (возможно, с ограничением: xsdRestriction).

A.6.2 category

Категория, связанная с этим объектом (необязательное значение).

A.6.3 code

Согласованный код структуры взаимодействия для этого объекта.

Пример -

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

A.6.4 dateLaMu

Дата и время последнего изменения этого объекта.

Пример -

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

A.6.5 description

Описание объекта.

Пример -

```
<... id="Doorleaf" >
…
<description>Створка плоской двери.</description>
…
</...>
```

A.6.6 end Date

Дата и время окончания срока действия этого объекта.

Пример -

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

A.6.7 firstMessage

Необязательное Булево значение, показывающее, разрешается ли использовать этот объект типа MessageInTransactionType как стартовое сообщение для новой транзакции. Значение по умолчанию - false.

Пример -

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

A.6.8 format

Разметка элемента (необязательная).

A.6.9 helpInfo

URL-адрес или URI-код ссылки на дополнительную информацию об этом объекте.

Пример -

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

A.6.10 initiatorToExecutor

Булево значение, представляющее направление, в котором сообщение должно посылаться.

Пример -

```
<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="..." >
```

Предполагается, что сообщение *TenderAcceptance* будет отправлено от клиента (исполнитель транзакции) исполнителю работ (инициатор).

A.6.11 interfaceType

Интерфейс типа или представление этого SimpleElementType для конкретного сообщения.

A.6.12 language

Язык для использования после продвижения объекта.

Пример -

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

A.6.13 namespace

Имя целевого пространства имен для идентификации сообщений, принадлежащих этой структуре взаимодействия.

Пример -

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

Примечание - Элемент **namespace**. Ранее все схемы структуры взаимодействия ссылались на одно целевое пространство имен, которое совпадало с пространством имен самой схемы взаимодействия. С помощью этого элемента в каждой схеме взаимодействия можно указать ее собственное пространство имен.

A.6.14 openSecondaryTransactionsAllowed

Необязательное логическое значение, которое позволяет продолжить основную транзакцию, даже если завершены не все подчиненные транзакции. Значение TRUE интерпретируется как сигнал, разрешающий продолжение (шлюз OR). Значение FALSE интерпретируется как сигнал, запрещающий продолжение, пока не будут завершены все подчиненные транзакции (шлюз AND). Если элемент openSecondaryTransactionsAllowed не указан, его значение интерпретируется как TRUE.

A.6.15 received

Логическое значение, показывающее, что предыдущее сообщение получено.

A.6.16 requiredNotify

С этим элементом не связана никакая семантика.

A.6.17 responsibilityFeedback

Ожидаемая обратная связь в направлении к другим ролям (необязательная строка).

A.6.18 responsibilityScope

Область действия/рамки определенных отношений ответственности роли (необязательная строка).

A.6.19 responsibilitySupportTask

Задачи, которые должны выполняться для поддержки других ролей. Например, делегированные ответственности (необязательная строка).

A.6.20 responsibilityTask

Задачи, возникающие из-за отношений ответственности данной роли (необязательная строка).

A.6.21 result

Ожидаемый результат выполнения транзакции.

A.6.22 send

Логическое значение, показывающее, было ли отправлено сообщение.

A.6.23 startDate

Дата и время начала срока действия объекта.

Пример -

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

A.6.24 state

Состояние видимости объекта, возможные значения:

Пример -

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

и

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

A.6.25 userLaMu

Пользователь последнего изменения объекта.

Пример -

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

A.6.26 valueList

Разделяемый точкой с запятой список значений, которые может принимать образец на уровне сообщения. Исходно этот элемент предназначался для поддержки перечислений. Сейчас это реализуется с помощью типа элемента UserDefinedType и элемента xsdRestriction. Значения перечисления задаются в xsdRestriction. С этим элементом больше не связывается никакая семантика.

Пример -

```
<SimpleElementType id="..." >
…
<valueList>Groen;Rood;Oker Geel</valueList>
…
```

</SimpleElementType>

A.6.27 xsdRestriction

Этот элемент указывает ограничение, которое будет выполняться на уровне сообщения в типе SimpleElementType этого типа UserDefinedType.

A.7 Ссылки

A.7.1 appendixTypes

Набор доступных для выбора типов дополнения.

A.7.2 complexElements

Ссылка на набор типов SimpleElementType (собранных в ComplexElementType).

Пример -

```
<... id="Abc" >
…
<complexElements>
<ComplexElementType id="ElementsSet1" >
...
<elements>
<SimpleElementType id="Element_A" >
…
</SimpleElementType>
<SimpleElementType id="Element_B" >
…
</SimpleElementType>
</elements>
</ComplexElementType>
<ComplexElementType id="ElementsSet2" >
…
</ComplexElementType>
<ComplexElementType id="ElementsSet3" >
...
</ComplexElementType>
</complexElements>
</...>
```

На уровне сообщения это конкретизируется в:

```
<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

Тип SimpleElementType может состоять из набора типов SimpleElementType (собранных в типе ComplexElementType).

Пример -

```
<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>
```

На уровне сообщения это конкретизируется в:

```
<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

Состояние для типа SimpleElementType, который будет использоваться в типе MessageType.

Пример -

```
<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>
```

В этом примере показано, что в типе MessageType со значением *M* объект *Doorheight* должен иметь значение не меньше 2000, хотя это не применяется на уровне типа SimpleElementType.

A.7.5 elements

Набор типов SimpleElementType, доступный внутри типа ComplexElementType.

Пример -

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

На уровне сообщения это конкретизируется в:

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

```
<door>
<Doorleaf>...</Doorleaf>
<Fastenings>...</Fastenings>
<UpperWindow>...</UpperWindow>
</Door>
</door>
</...>
```

Обязательным является то, что все элементы точно указываются только один раз. У этих дверей всегда имеется верхнее окно. Число дверей не ограничено.

A.7.6 executor

Роль 'исполняющий' (RoleType), связанная с конкретной транзакцией.

Пример -

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

A.7.7 group

Тип GroupType, связанный с сообщением в пределах конкретной транзакции.

Пример -

```
<MessageInTransactionType id="..." >
…
<message>
<MessageType id="M" >
…
</MessageType>
</message>
<group>
<GroupType id="G" >
…
</GroupType>
</group>
<transaction>
<TransactionType id="T" >
…
</TransactionType>
</transaction>
…
</MessageInTransactionType>
```

Сообщение M в транзакции T принадлежит к группе G (может существовать несколько сообщений M в транзакции T, принадлежащих к той же самой или другой группе).

A.7.8 Initiator

Роль 'инициирующий' (RoleType), принадлежащая к конкретной транзакции.

Пример -

```
<TransactionType id="TenderTraject" >
…
```

```
<initiator>
<RoleType id="Performer" >
…
</RoleType>
</initiator>
…
</TransactionType>
```

A.7.9 message

Сообщение в MessageInTransactionType.

Пример -

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

A.7.10 previous

Тип MessageInTransactionType может обеспечивать принудительное выполнение предыдущего сообщения в конкретной транзакции (этот предыдущий тип MessageInTransactionType не должен принадлежать к тому же самому типу TransactionType).

Пример -

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

Транзакции, которые могут выполняться в данной транзакции.

Пример -

```
<TransactionType id="AcguireWork" >
…
<subTransactions>
<TransactionType id="AcguisitionTrack" >
…
</TransactionType>
<TransactionType id="TenderTrack" >
```

…
</TransactionType>
</subTransactions>
</TransactionType>

Тип TransactionType со значением *AcquireWork* состоит из типов TransactionType со значениями *AcquisitionTrack* и *TenderTraject*.

A.7.11 simpleElements

Набор простых типов элементов, принадлежащих к данному сложному типу элементов.

Пример -

```
<ComplexElementType id="Door" >
…
<simpleElements>
<SimpleElementType id="Doorleaf" >
…
</SimpleElementType>
<SimpleElementType id="HingesAndLocks" >
…
</SimpleElementType>
<SimpleElementType id="UpperWindow" >
…
</SimpleElementType>
</simpleElements>
</ComplexElementType>
```

На уровне сообщения это может привести к следующему результату:

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

A.7.12 subTransactions

Транзакции, которые могут быть запущены из данной транзакции.

Пример -

```
<TransactionType id="AcguisitionOfWork" >
…
<subTransactions>
<TransactionType id="AcguisitionTraject" >
…
</TransactionType>
<TransactionType id="TenderTraject" >
…
</TransactionType>
</subTransactions>
```

```
</TransactionType>
```

A.7.13 transaction

Транзакция в типе MessageInTransactionType.

Пример -

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

A.7.14 transactionPhase

Объект TransactionPhase для конкретного типа MessageType в конкретном типе TransactionType.

Пример -

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

Где тип MessageType, имеющий значение M внутри конкретного типа TransactionType, имеющего значение T, определяет тип TransactionPhaseType, имеющий значение TP.

A.7.15 userDefinedType

Ссылка на тип UserDefinedType, который задает формат типа SimpleElementType.

Пример -

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

Тип SimpleElementType Height задает формат Integer для каждого образца (возможно, с ограничением в виде

xsdRestriction).

Приложение В
(обязательное)

Определение шаблонов

В.1 Введение

После того как актуальная структура взаимодействия определена, потребуется файл схемы взаимодействия, содержащий правила для фактической коммуникации. Правила, которые не зависят от фактической коммуникации, включены в файл шаблонов. В этом приложении дано определение шаблонов в формате языка EXPRESS. Кроме того, приведены описания типов элементов, атрибутов, элементов и ссылок.

В.2 Определение шаблонов[1)]

_____

1) В настоящем стандарте приводится определение шаблонов, представленное в ИСО 29481-2.

```
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 Шаблоны

B.3.1 AppendixGroup

Атрибуты: id

Элементы: state, dateLaMu, userLaMu

Ссылки: group

Таблица отображения для отношений n:m между дополнениями и группами

Пример - на уровне сообщения:

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

```
</AppendixGroup>
```

Ассоциированная часть структуры взаимодействия:

```
<GroupType id="StandardGroupType" >
<description>Стандартная группа</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

Атрибуты: id

Элементы: name, fileLocation, fileType, fileVersion, documentIdentification, documentVersion, documentReference, objectCode, startDate, endDate, state, dateLaMu, userLaMu, language

Ссылки: appendixGroup, message

Регистрация присоединенных файлов.

Пример - на уровне сообщения:

```
<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>
```

Ассоциированная часть структуры взаимодействия:

```
<description>Стандартное определение приложения (без определенных пользователем полей данных</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

Атрибуты: id

B.3.4 GroupTemplate

Атрибуты: id

Элементы: name, description, creationDate, startDate, endDate, state, dateLaMu, userLaMu, versionNo

Ссылки: transaction

Группирование дополнений сообщения для восстановления ассоциированных документов.

Пример - на уровне сообщения:

```
<StandardGroupType id="MenuBackgrounds" >
<name>Menu Pictures</name>
<description>Набор фоновых картинок для оформления меню</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>
```

Ассоциированная часть структуры взаимодействия:

```
<GroupType id="StandardGroupType" >
<description>Стандартная группа</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>Транзакция для получения соответствующего меню</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="Сотрудник"/ >
</executor>
</TransactionType>
```

B.3.5 MessageInTransactionTemplate

Атрибуты: id

Элементы: identification, dateSend, dateRead, state, dateLaMu, userLaMu, initiatorToExecutor

Содержит фактическое значение MessageInTransactionType в сообщении для извлечения состояния рабочего процесса транзакции.

B.3.6 MessageTemplate

Атрибуты: id

Элементы: identification, dateSend, dateRead, state, dateLaMu, userLaMu, initiatingTransactionMessageID, initiatorToExecutor

Ссылки: transaction, messageInTransaction, template

Экземпляр типа MessageType. Эта сущность осуществляет фактический обмен информацией между объектами OrganisationTemplate (организациями).

Пример - на уровне сообщения:

```
<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>
```

Ассоциированная часть структуры взаимодействия:

```
<TransactionType id="MenuObtainingTransaction" >
<description>Транзакция для получения соответствующего меню</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="Сотрудник"/ >
</executor>
</TransactionType>
<MessageType id="HandOutOfMenuMessage" >
<description>Сообщение, содержащее меню.</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>Доступные пункты меню</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

Атрибуты: id

Элементы: name, abbreviation, state, dateLaMu, userLaMu

Ссылки: contactPerson, template

Организация, участвующая в проекте по инициализации или исполнению TransactionTemplate (транзакции).

Пример - на уровне сообщения:

<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>

Ассоциированная часть структуры взаимодействия:

<PersonType id="StandardPersonType" >
<description>Тип стандартного участника процесса</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>Тип стандартного подразделения</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

Атрибуты: id

Элементы: state, dateLaMu, userLaMu

Ссылки: organization, contactPerson, role, substituting, successor

Пользователь, выполняющий определенную роль для некоторой организации.

Пример - на уровне сообщения:

```
<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>
```

Ассоциированная часть структуры взаимодействия:

```
<PersonType id="StandardPersonType" >
<description>Тип стандартного участника процесса</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>Тип стандартного подразделения</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>Потребитель</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

Атрибуты: id

Элементы: userName, name, state, dateLaMu, userLaMu

Лицо, участвующее в проекте путем выполнения некоторой роли или как контактное лицо в конкретной

организации.

Пример - на уровне сообщения:

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

Ассоциированная часть структуры взаимодействия:

```
<PersonType id="StandardPersonType" >
<description>Тип стандартного участника процесса</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>
```

B.3.10 ProjectTemplate

Атрибуты: id

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

Проект, который должен соответствовать этой структуре.

Пример - на уровне сообщения:

```
<StandardProjectType id="IDM2" >
<name>The project IDM2</name>
<description>Формализация систематики IDM2</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>
```

Ассоциированная часть структуры взаимодействия:

```
<ProjectType id="StandardProjectType" >
<description>Тип стандартного проекта</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

Атрибуты: id

Элементы: name, description, state, dateLaMu, userLaMu, category

Роль, выполняемая от имени организации пользователем PersonTemplate (лицом).

Пример - на уровне сообщения.

```
<Consumer id="Customer" >
<name>Role as customer</name>
```

<description>Роль в качестве клиента</description>
<state>active</state>
dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
<userLaMu>bapa</userLaMu>
</Consumer>

Ассоциированная часть структуры взаимодействия:

<RoleType id="Consumer" >
<description>Потребитель</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

Атрибуты: id

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

Ссылки: transaction

Фактическая фаза транзакции.

Пример - на уровне сообщения:

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

Ассоциированная часть структуры взаимодействия:

<TransactionType id="ObtainMenuTransaction" >
<description>Транзакция для получения соответствующего меню</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>Меню запрошено, но еще не дано/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

Атрибуты: id

Элементы: number, name, description, startDate, endDate, state, dateLaMu, userLaMu, result

Ссылки: initiator, executor

Транзакция, активирующая MessageTemplates (сообщения), которые должны участвовать в обмене для выполнения некоторых задач.

Пример - на уровне сообщения:

```
<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>
```

Ассоциированная часть структуры взаимодействия:

```
<TransactionType id="ObtainMenuTransaction" >
<description>Транзакция для получения соответствующего меню</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="Сотрудник"/ >
</executor>
</TransactionType>
```

B.4 Атрибуты

B.4.1 Id

Идентификатор. Уникальный код для идентификации экземпляра объекта в сообщении.

Примечание - Значение идентификатора должно быть уникальным в XML-документе. Поэтому простой серийный номер уже будет удовлетворять этому условию. Следовательно, не существует убедительных причин для ограничения значений идентификаторов только глобальными уникальными идентификаторами (GUID) (хотя в большинстве реализаций действительно применяются глобальные уникальные идентификаторы). Стандарт XML (http:/www.w3.org/XML/) требует, чтобы идентификатор начинался с буквы или с символа '_'.

B.5 Элементы

B.5.1 Abbreviation

Сокращенное названия организации. Этот элемент будет использоваться в качестве префикса номера транзакции для идентификации транзакции.

Примечание - Элемент **abbreviation** (сокращение). Элемент abbreviation содержит краткое название организации. Одно из его применений - пометить транзакции (в сочетании с порядковым номером), запущенные данной организацией, чтобы облегчить неформальную коммуникацию между пользователями-участниками.

B.5.2 category

Категория, позволяющая классифицировать данный экземпляр объекта.

Пример - на уровне сообщения:

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

B.5.3 creation Date

Дата создания конкретного экземпляра объекта в данном сообщении.

Пример - на уровне сообщения:

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

B.5.4 dateLaMu

Дата последнего изменения.

Пример - на уровне сообщения:

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

B.5.5 dateReached

Дата "dateReached" - это атрибут даты/времени элемента "TransactionPhaseTemplate". Он представляет собой дату/время достижения конкретной фазы.

Пример - на уровне сообщения:

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

B.5.6 dateRead

Дата чтения данного сообщения.

Пример - на уровне сообщения:

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

B.5.7 dateSend

Дата отправки данного сообщения.

Пример - на уровне сообщения:

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

B.5.8 description

Описание данного экземпляра объекта.

Пример - на уровне сообщения:

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

B.5.9 documentIdentification

Уникальный номер документа или ссылка на документ или файл для идентификации документа.

Примечание - Настоящий стандарт предоставляет свободу в установлении различий между версиями документа и версиями файла (см. раздел B.5.13).

B.5.10 documentReference

Ссылка для идентификации файла или документа.

B.5.11 documentVersion

Версия документа или файла.

B.5.12 endDate

Дата истечения срока действия конкретного экземпляра объекта в данном сообщении.

Пример - на уровне сообщения:

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

B.5.13 fileLocation

Местоположение файла, предпочтительно адрес в Интернете или путь к серверу общего доступа.

Пример - на уровне сообщения:

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

Примечание - Настоящий стандарт предоставляет свободу в установлении различий между версиями документа и версиями файла (см. раздел B.5.9).

B.5.14 fileType

Тип файла, предпочтительно тип MIME (Multipurpose Internet Mail Extensions).

Пример - на уровне сообщения:

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

B.5.15 fileVersion

Версия файла - возрастающее целое число или дата.

Пример - на уровне сообщения:

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

B.5.16 identification

Ориентированная на пользователя идентификация данного сообщения.

Пример - на уровне сообщения:

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

B.5.17 initiatingTransactionMessageID

Ссылка на сообщение, принадлежащее к подчиненной транзакции, которая инициировала это сообщение.

B.5.18 initiatorToExecutor

Направление коммуникации данного конкретного сообщения.

Пример - на уровне сообщения:

```
<ExampleMessage id="..." >
…
<initiatorToExecutor>false</initiatorToExecutor>
…
<transaction>
<ExampleTransaction id="..." >
…
```

```
<initiator>
<PersonInRole id="A" >
…
</PersonInRole>
</initiator>
<executor>
<PersonInRole id="B" >
…
</PersonInRole>
</executor>
</ExampleTransaction>
</transaction>
…
</ExampleMessage>
```

В данном примере ExampleMessage будет передаваться от B (исполнитель) к A (инициатор).

B.5.19 language

Язык содержимого дополнения.

Пример - на уровне сообщения:

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

B.5.20 name

Наименование (STRING)

B.5.21 number

Номер транзакции.

Примечание - Элемент **number.** Порядковый номер транзакции (см. также предыдущий элемент).

B.5.22 objectCode

Возможность связи с внешним индексом. Например, структура декомпозиции работ, пакеты работ или спецификации зданий.

B.5.23 result

Результат данной транзакции.

B.5.24 startDate

Начальная дата для конкретного экземпляра объекта в данном сообщении.

Пример - на уровне сообщения:

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

B.5.25 state

Текущее состояние данного экземпляра объекта.

Пример - на уровне сообщения:

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

B.5.26 userLaMu

Пользователь, выполнивший последнее изменение

Примечание - Это не ссылка на создание экземпляра PersonTypeInstance.

Пример - на уровне сообщения:

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

B.5.27 userName

Инициалы пользователя, например, триграмма (комбинация из трех символов).

Пример - на уровне сообщения:

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

B.5.28 versionNo

Номер версии экземпляра объекта, подключенного к данному проекту; после изменения данного экземпляра объекта это поле должно быть также обновлено.

Пример - на уровне сообщения:

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

B.6 Ссылки

B.6.1 AppendixGroup

Подраздел для идентификации некоторого дополнения.

Пример - на уровне сообщения:

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

…
</Appendix>

Считается, что в ассоциированной структуре определен тип AppendixType, имеющий значение Appendix.

B.6.2 contactPerson

Лицо, связанное с объектом PersonInRole или связанное с конкретной организацией.

Пример - на уровне сообщения:

Organisation id="..." >
…
<contactPerson>
<Person id="..." >
…
</Person>
</contactPerson>
</organisation>

Считается, что в ассоциированной структуре определен тип OrganisationType, имеющий значение Organization, и тип PersonType, имеющий значение Person.

B.6.3 Executor

Объект PersonInRole, который будет действовать как исполнитель в этой транзакции.

B.6.4 Group

Общая группа для сборки набора дополнений.

Пример - на уровне сообщения:

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

Считается, что в ассоциированной структуре определен тип GroupType, имеющий значение Group.

B.6.5 Initiator

Объект PersonInRole, который будет действовать как инициатор в этой транзакции.

B.6.6 Message

Сообщение, содержащее конкретное дополнение.

Пример - на уровне сообщения:

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

Считается, что в ассоциированной структуре определен тип AppendixType, имеющий значение Appendix, и тип

MessageType, имеющий значение Message.

B.6.7 messageInTransaction

Ссылка на положение этого сообщения в потоке транзакции.

B.6.8 Organization

Организация, принадлежащая объекту PersonInRole.

Пример - на уровне сообщения:

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

Считается, что в ассоциированной структуре определен тип OrganisationType, имеющий значение Organization.

B.6.9 Role

Ссылка на роль, выполняемую от имени организации пользователем PersonTemplate (лицом).

B.6.10 Substituting

Объект PersonInRole, авторизованный для отправки сообщений от имени данного лица.

Примечание - Ссылка **substituting**. Ссылка на формальный объект PersonInRole для данной транзакции, позволяющая действовать другому лицу в отсутствие лица, реализующего формальный объект PersonInRole. Оба лица должны ссылаться на один и тот же тип роли.

B.6.11 Successor

Преемник другого лица в некоторой роли.

B.6.12 Transaction

Транзакция, содержащая конкретную группу, фазу транзакции или конкретное сообщение.

Пример - на уровне сообщения:

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

Считается, что в ассоциированной структуре определен тип MessageType, имеющий значение Message, и тип TransactionType, имеющий значение Transaction.

Приложение С
(справочное)

Пример карты взаимодействия для упрощенного конструкторского бюро

С.1 Общие положения

В приведенном примере создания структуры взаимодействия область действия взаимодействия ограничена упрощенным взаимодействием в пределах конструкторского бюро (см. пример в разделе 4). Взаимодействие определяется в структуре путем объявления:

- ролей,

- транзакций,

- сообщений в транзакциях,

- порядка сообщений в транзакции,

- элементов данных в сообщениях.

С.2 Роли и транзакции

Для иллюстрации области действия для взаимодействия в данном примере действия коммуникации изображены на 'карте взаимодействия', отображающей роли, связанные с транзакциями.

Роли:

- CR1: Client; (клиент)

- R1: Project leading; (управление проектом)

- R2: System engineering; (системное проектирование)

- R3: 3D engineering; (проектирование трехмерных моделей)

- R4: Cost engineering (стоимостное проектирование).

Сводка взаимодействий приведена в таблице ролей в транзакциях.

Таблица С.1 - Таблица ролей в транзакциях для упрощенного конструкторского бюро

Тип транзакции	Инициирующая роль	Исполняющая роль
T1 Разработка проекта	CR1 Клиент	R1 Управление проектом
T2 Разработка спецификации системы	R1 Управление проектом	R2 Системное проектирование
T3 Разработка трехмерной модели	R1 Управление проектом	R3 Проектирование трехмерных моделей
T4 Разработка сметы	R1 Управление проектом	R4 Стоимостное проектирование

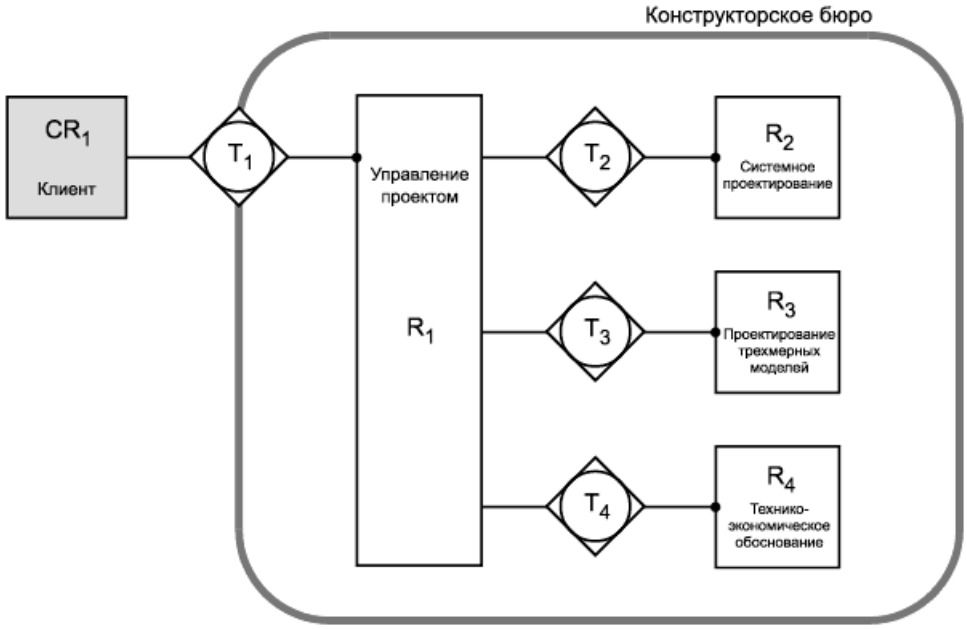

Рисунок С.1 - Карта взаимодействия, определяющая соответствующие типы ролей и типы транзакций

С.2.1 RoleType

Роли определены в структуре как 'RoleTypes'. Каждый RoleType может быть определен следующими терминами:

- идентификатор (идентификация roleType),

- [Должен иметь допустимое значение идентификатора XML (квалифицированное имя). Существует несколько ограничений: * должно быть уникальным в модели, * некоторые символы не разрешены ('/', '$', '#', '@', ...), * без пробелов, * не должно начинаться с цифры]

- описание (спецификация roleType),

- ответственности (область действия, задача, задача поддержки и обратная связь),

- метаданные (последнее изменение, справочная информация и т.д.).

Следующие четыре роли данного примера определены в структуре (см. приведенный ниже пример в представлении XML):

<RoleType id="CR1_Client" >
<description>CR1: **Клиент**</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>Несет ответственность за реализацию проекта, выражаемую в терминах времени, качества и стоимости </responsibilityScope>

<language></language>
<category></category>
<helpInfo></helpInfo>
<code></code>
<responsibilityScope>Должен доставить проект в соответствии с требованиями к контракту
</responsibilityScope>
<responsibilityTask></responsibilityTask>
<responsibilitySupportTask></responsibilitySupportTask>
<responsibilityFeedback></responsibilityFeedback>
</RoleType>
<RoleType id="R2_System_Engineering" >
<description>**R2: Системное проектирование**</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>Должен доставить спецификацию системы в соответствии с контрактными соглашениями/responsibilityScope>
<responsibilityTask></responsibilityTask>
<responsibilitySupportTask></responsibilitySupportTask>
<responsibilityFeedback></responsibilityFeedback>
</RoleType>
<RoleType id="**R3_3D_Engineering**" >
<description>**R3: Проектирование трехмерных моделей**</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>Должен доставить 3D-модель в соответствии с контрактными соглашениями/responsibilityScope>
<responsibilityTask></responsibilityTask>
<responsibilitySupportTask></responsibilitySupportTask>
<responsibilityFeedback></responsibilityFeedback>
</RoleType>
<RoleType id="R4_Cost_Engineering" >
<description>**R4: Стоимостное проектирование**</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>Должен доставить расчет стоимости в соответствии с контрактными соглашениями/responsibilityScope>
<responsibilityTask></responsibilityTask>
<responsibilitySupportTask></responsibilitySupportTask>
<responsibilityFeedback></responsibilityFeedback>
</RoleType>

C.3 Сообщение в транзакции

Транзакция содержит набор сообщений, обмен которыми выполняется для какой-либо определенной цели. Транзакция также предусматривает участвующие роли, точку в жизненном цикле и последовательность, в которой должны доставляться сообщения (если это необходимо).

C.3.1 TransactionType

Транзакции определяются в структуре как 'TransactionTypes'. Каждый TransactionType может быть определен следующими терминами:

- идентификатор (идентификация TransactionType),

- описание (спецификация TransactionType),

- метаданные (результат применения TransactionType, справочная информация и т.д.),

- типы RoleType, которые включены в TransactionType.

Следующие четыре типа TransactionType в данном примере изображены в приведенном ниже представлении XML:

- T1: Обмен проектом;

- T2: Обмен системной спецификацией;

- T3: Обмен 3D-моделью;

- T4: Обмен расчетом стоимости.

<TransactionType id="T1" >
<description>T1 Доставка проекта </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>Проект доставлен</result>
<initiator>
<RoleTypeRef idref="CR1_Client"/ >
</initiator>
<executor>
<RoleTypeRef idref="R1_Project_Leading"/ >
</executor>
</TransactionType>
<TransactionType id="T2" >
<description>T2 Доставка спецификации ситстемы</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:>Спецификация системы доставлена</result>
<initiator>
<RoleTypeRef idref="R1_Project_Leading"/ >
</initiator>
<executor>
<RoleTypeRef idref="R2_System_Engineering"/ >
</executor>
</TransactionType>
<TransactionType id="T3" >
<description>T3 Доставка 3D-модели</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-модель доставлена</result>
<initiator>
<RoleTypeRef idref="R1_Project_Leading"/ >
</initiator>
<executor>
<RoleTypeRef idref="R3_3D_Engineering"/ >
</executor>
</TransactionType>
<TransactionType id="T4" >
<description>T4 Доставка расчета стоимости</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>Расчет стоимости доставлен</result>
<initiator>
<RoleTypeRef idref="Rl_Project_Leading"/ >
</initiator>
<executor>
<RoleTypeRef idref="R4_Cost_Engineering"/ >
</executor>
</TransactionType>
```

IDM выделяет роль, делающую запрос (инициатор), и роль, обеспечивающую осуществление этого запроса (исполнитель). В каждой транзакции может быть только одна инициирующая роль и одна исполняющая роль. Например, тип TransactionType, имеющий значение 'T3 Deliver 3D model' (Обмен 3D-моделью), может быть инициирован только ролью R1 Project leading (Управление проектом) и выполнен инженером - разработчиком трехмерных моделей в роли R3.

С помощью транзакций передача информации осуществляется в контексте процесса.

C.3.2 TransactionPhaseType

С типами TransactionType могут быть связаны фазы, позволяющие определить состояние взаимодействия внутри транзакции. В сущности, типы TransactionPhaseType в TransactionType представляют собой массив транзакций. Определения TransactionPhaseType обеспечивают идентификацию типов TransactionPhaseType и некоторых метаданных. Число фаз и их описание не ограничены.

В данном примере определены следующие типы TransactionPhaseType, соотносящиеся с рисунком 2 в разделе 3:

- желательный результат;

- результат запрошен;

- результат обещан;

- результат получен;

- результат объявлен;

- результат принят.

В приведенном ниже примере показано определение TransactionPhaseType в представлении XML:

```
<TransactionPhaseType id="Desired_Result" >
<description>Желательный результат</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>Результат принят</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>Результат получен</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>**Результат обещан**</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>**Результат запрошен**</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>**Результат объявлен**</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

Сообщение представляет собой заполненную информационную модель, содержащую данные. MessageType определяет структуру и указывает элементы для содержимого сообщения.

В каждом сообщении имеются группы сегментов (составных элементов). Эти составные элементы определяются типами ComplexElementType. В каждом сегменте может находиться совокупность составных элементов и/или одиночных элементов. Одиночные элементы определяются типами SimpleElementType.

В приведенном ниже примере показано определение типов MessageType в представлении XML, изображенное графически на рисунке 5 в разделе 3:

- запрос 3D-модели;

- задание выполнено, запрос одобрения;

- запрос корректировок;

- задание утверждено;

- задание не утверждено.

<MessageType id="Request_for_3D_model" >
<description>**Запрос 3D-модели**</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**"/ >
<ComplexElementTypeRef idref="**RemarksComplexElement**"/ >
<ComplexElementTypeRef idref="**DeadlineComplexElement**"/ >
</complexElements>
</MessageType>
<MessageType id="Work_done_and_request_approval" >
<description>**Задание выполнено, запрос одобрения** </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="**WorkSpecificationComplexElenient**"/ >
</complexElements>
</MessageType>
<MessageType id="Request_adjustments" >
<description>**Запрос корректировок**</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>**Задание утверждено**</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>**Задание не утверждено**</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 также используется для идентификации сообщений в транзакции. Отношение между messageType и transactionType определяется в другом типе: MessageInTransactionType.

C.3.4 MessageInTransactionType

Транзакция содержит набор сообщений, обмен которыми выполняется для какой-либо цели. Определения MessageInTransactionType обеспечивают связывание MessageTypes с TransactionTypes. Один и тот же тип MessageType может быть связан со всеми определенными типами TransactionType. Также можно связать один и тот же MessageType более одного раза (даже неограниченное число раз) с тем же самым TransactionType.

Каждый MessageInTransactionType может быть определен следующими терминами:

- идентификатор (идентификация MessageInTransactionType),

- направление сообщения, которое должно быть отправлено (от инициатора к исполнителю = true; от исполнителя к инициатору = false),

- метаданные MessageInTransactionType,

- ссылка на тип MessageType (положение которого определено в типе MessageInTransactionType),

- ссылка на тип (или типы) TransactionType для связывания с типами MessageType, на которые дана ссылка,

- ссылка на TransactionPhase (фазу), в состояние которой переходит процесс (после отправки MessageType в типе TransactionType, как это было определено типом MessageInTransactionType),

- ссылка на тип GroupType.

В приведенном ниже примере показано определение в представлении XML двух типов MessageInTransactionType из примера запроса трехмерной модели. Пример будет понятен, если ознакомиться с рисунком 5 (в ISO 29481-2 IDM, часть 2 Карта взаимодействия; глава 3).

Приведенный ниже пример в представлении XML показывает, что TransactionType T3 запускается с помощью MessageType 'Запроса трехмерной модели'. Это первое сообщение в транзакции, поскольку не существует

предыдущих типов MessageInTransactionType, определенных в MessageInTranactionType 'MessageInTransaction1'. Направление MessageInTransaction1 указано как "от инициатора к исполнителю" (значение = true).

Второй тип MessageInTransactionType, определенный как 'MessageInTransaction2', ссылается на тип MessageType 'Задание выполнено, запрос одобрения'. Рисунок 5 показывает, что это сообщение должно следовать за двумя предыдущими сообщениями:

- Запроса трехмерной модели;

- Запрос корректировок.

Эти два предыдущих сообщения определены как типы MessageInTransactionType (номер 1 и номер 3) в MessageInTransactionType2. Направление MessageInTransaction2 указано как "от исполнителя к инициатору" (значение = false).

```
<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

К сообщениям могут прилагаться вложения. В виде вложения может передаваться требование к обмену исполняющей роли, а результат (вклад в BIM) будет доставлен инициирующей роли.

Определения AppendixType предназначены для ограничения информационных элементов компонентов, связанных с документами, путем добавления метаданных в документы, вложенные в сообщение.

В приведенном ниже представлении XML показано определение AppendixType. Метаданные определяются типами SimpleElementType в ComplexElementType 'AppendixComplexElement'.

```
<AppendixType id="StandardAppendixType" >
<description>Стандартное дополнение</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 Ограничение содержимого сообщений

Третьим и последним шагом определения структуры взаимодействия является определение компонентов в типах MessageType с целью:

- структурировать типы MessageType;

- ограничить входные данные содержимого.

Определение MessageType уже содержит составные элементы сообщения. Эти составные элементы определены как типы ComplexElementType.

Типы ComplexElementType состоят из типов SimpleElementType (или, возможно, также из других типов ComplexElementType). Типы SimpleElementType связываются с типами UserDefinedType для определения ограничений входных данных содержимого. Эта структура графически изображена ниже.

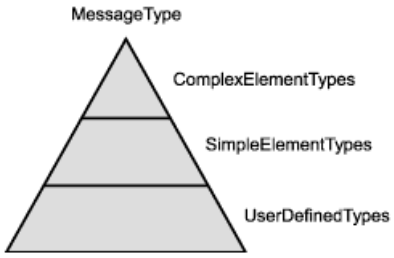

Рисунок С.2 - Компоненты моделирования сообщения

C.3.7 Типы ComplexElementType

Определения ComplexElementType обеспечивают:

- ограничение информационных элементов компонентов,

- используемых механизмами иерархия определений типов* (в XML) для получения сложного типа из другого простого или сложного типа,

_____

* Текст документа соответствует оригиналу. - Примечание изготовителя базы данных.

- определения вклада в проверку соответствия схемы документу (post-schema-validation infoset) для элементов,

- ограничения возможности получения дополнительных типов из заданных сложных типов,

- управления разрешением на подстановку в образце элементов полученного типа для элементов, объявленных в модели содержимого как элементы заданного сложного типа.

Если обратиться к приведенному примеру использования MessageType 'Запрос трехмерной модели', связанного с транзакцией Т3 (Запрос трехмерной модели), этот MessageType содержит следующие типы ComplexElementType:

- resultRequestedComplexElement;

- workIdentificationComplexElement;

- workSpecificationComplexElement;

- remarksComplexElement;

- deadlineComplexElement.

Тип ComplexElementType 'WorkSpecification' показан ниже в представлении XML. ComplexElementType определяет его идентификацию, ссылку на простые элементы, которые передают содержимое в ComplexElementType, и его метаданные.

В XML-представлении для ComplexElementType 'WorkSpecificationComplexElement' показано, что ComplexElementType содержит два типа SimpleElementType:

- ScopeOfWork;

- CostEstimation.

```
<ComplexElementType id="WorkSpecificationComplexElement" >
<description>Спецификация задания</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>
```

С.3.8 Типы SimpleElementType

Определения типа SimpleElementType обеспечивают:

- определение его идентификации,

- задание 'пространства значений' и ограничений входных данных.

В приведенном ниже представлении XML показано определение одного типа SimpleElementType: CostEstimation. Этот тип SimpleElementType является частью типа ComplexElementType 'WorkSpecificationComplex ElementType' (из предыдущего примера).

<SimpleElementType id="**CostEstimation**" >
<description>Оценка стоимости</description>
<interfaceType/>
<state>active</state>
<dateLaMu>2010-12-10T12:36:02.527+01:00</dateLaMu>
<userLaMu>Ronald</userLaMu>
<language/>
<category/>
<helpInfo>Информация о затратах </helpInfo>
<valueList/>
<userDefinedType>
<UserDefinedTypeRef idref="EURO"/ >
</UserDefinedType>
</SimpleElementType>

C.3.9 Типы UserDefinedType

Типы UserDefinedType используются для определения ограничений для входных данных содержимого (путем определения типов dataType). Все возможные ограничения XML разрешено использовать для определения типов UserDefinedType в структуре взаимодействия. В данном примере использованы только типы данных 'STRING' и 'EURO'.

В приведенном ниже примере показано в представлении XML определение типа UserDefinedType, обеспечивающего:

- идентификацию,

- метаданные,

- baseType,

- ограничение XSD (для более конкретного определения ограничения или для определения типов перечислений).

- <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>Суммы в евро, указанные с двумя десятичными знаками</helpInfo>
</UserDefinedType>
- <UserDefinedType id="**STRING**" >
<description>**Строки символов**</description>
<state>**active**</state>
<dateLaMu>**2010-10-01T00:00:00**</dateLaMu>
<userLaMu>**Ronald**</userLaMu>
<baseType>**STRING**</baseType>
<xsdRestriction />
<language />
<helpInfo>Тип данных String представляет строки символов в XML. 'Пространство значений' строки - это набор последовательностей символов конечной длины.</helpInfo>
</UserDefinedType>

C.3.10 Заключение

Были определены все типы ElementType, достаточные для создания структуры взаимодействия. Для проверки правильности необходимо проверить соответствие структуры ее схеме XSD. Следующим шагом должно быть продвижение структуры взаимодействия для получения схемы взаимодействия. Эти действия описаны в подразделе 4.8.

Приложение D
(справочное)

Основные идеи алгоритма промотора

D.1 Основные идеи алгоритма промотора

Алгоритм промотора превращает структуру взаимодействия в схему взаимодействия. Поскольку структура взаимодействия записывается в формате XML, а схема взаимодействия имеет формат XSD, можно заключить, что алгоритм промотора создает абстрактные типы для элементов данных структуры взаимодействия в структуре сущностей и связей в схеме взаимодействия.

Базовый алгоритм инструмента промотор преобразовывает все элементы данных () "*Type" структуры взаимодействия (RoleType, TransactionType, MessageType, ComplexElementType, SimpleElementType и т.д.) в элементы сложных типов XSD. Конечно, необходимо руководствоваться определенными правилами, гарантирующими, что результирующая схема XSD представляет собой корректную схему XML. Например, значение атрибута ID элемента данных *Type интерпретируется как наименование эквивалентного ему элемента сложного типа XSD. Следовательно, атрибут ID в структуре взаимодействия должен быть не чем-то, похожим на ID = "Role003", а более похожим на ID = "Project_Manager".

Схема XSD организована таким образом, что типы MessageType являются основными сложными элементами для структурирования содержимого сообщения на уровне фактического обмена в транзакции. Такое сообщение должно содержать достаточную информацию для отслеживания фактического положения этого сообщения в общем потоке сообщений и для определения последующих сообщений, которые могут быть отправлены из этого положения.

Для генерации правильных типов атрибутов и связанных типов для результирующей схемы XSD промотор обращается к файлу, описывающему, какой шаблон и для каких сложных типов элементов необходимо использовать (файл шаблонов).

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ISO 29481-1	IDT	ГОСТ Р 10.0.03-2019/ИСО 29481-1:2016 "Система стандартов информационного моделирования зданий и сооружений. Информационное моделирование в строительстве. Справочник по обмену информацией. Часть 1. Методология и формат"
Примечание - В настоящей таблице использовано следующее условное обозначение степени соответствия стандарта: - IDT - идентичный стандарт.		

Библиография

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

УДК 004.9:006.354

ОКС 91.010.01
35.240.67
35.240.01

Ключевые слова: система стандартов, информационное моделирование, здания и сооружения, строительство, справочник, обмен информацией, структура взаимодействия

Электронный текст документа
подготовлен АО "Кодекс" и сверен по:
официальное издание
М.: Стандартинформ, 2019