



НАЦІОНАЛЬНИЙ СТАНДАРТ УКРАЇНИ

---

**ДСТУ EN ISO 29481-2:20\_\_**  
**(EN ISO 29481-2:2016, IDT; ISO 29481-2:2012, IDT)**

**ІНФОРМАЦІЙНІ МОДЕЛІ БУДІВЕЛЬ.  
НАСТАНОВА З ДОСТАВЛЯННЯ ІНФОРМАЦІЇ.**

**Частина 2. Інтерактивне середовище**

*(Проект, остаточна редакція)*

Київ  
ДП «УкрНДНЦ»  
20\_\_

## ПЕРЕДМОВА

1. РОЗРОБЛЕНО: Технічний комітет стандартизації «Металобудівництво» (ТК 301), Товариство з обмеженою відповідальністю «Український інститут сталевих конструкцій імені В. М. Шимановського»
2. ПРИЙНЯТО ТА НАДАНО ЧИННОСТІ: наказ Державного підприємства «Український науково-дослідний і навчальний центр проблем стандартизації, сертифікації та якості» (ДП «УкрНДНЦ») від \_\_.\_\_\_\_. 20\_\_ р. № \_\_\_\_\_ з 20\_\_ \_\_ \_\_
3. Національний стандарт відповідає EN ISO 29481-2:2016 Building information models — Information delivery manual — Part 2: Interaction framework (ISO 29481-2:2012) (Інформаційні моделі будівель. Настанова з доставляння інформації. Частина 2. Інтерактивне середовище) і внесений з дозволу CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels, Belgium. Усі права щодо використання європейських стандартів у будь-якій формі й будь-яким способом залишаються за CEN  
  
Ступінь відповідності – ідентичний (IDT)  
  
Переклад з англійської (en)
4. Цей стандарт розроблено згідно з правилами, установленими в національній стандартизації України
5. УВЕДЕНО ВПЕРШЕ

---

**Право власності на цей національний стандарт належить державі.  
Заборонено повністю або частково видавати, відтворювати  
задля розповсюдження і розповсюджувати як офіційне видання  
цей національний стандарт або його частини на будь-яких носіях інформації  
без дозволу ДП «УкрНДНЦ» чи уповноваженої ним особи**

ДП «УкрНДНЦ», 20\_\_

## ЗМІСТ

	С.
Національний вступ .....	V
Передмова до ISO 29481-2:2012.....	VI
Вступ.....	VIII
1 Сфера застосування .....	1
2 Нормативні посилання.....	1
3 Терміни та визначення понять.....	2
4 Основоположні принципи .....	3
4.1 Загальні положення.....	3
4.2 BIM та IDM.....	3
4.3 Компоненти IDM .....	4
4.4 Основні принципи ділової комунікації .....	5
4.5 Карта взаємодії.....	7
4.6 Повідомлення у транзакції.....	9
4.7 Інтерактивне середовище.....	10
4.8 Підтримання програмних рішень.....	11
5 Формат структури взаємодії.....	14
5.1 Вступні положення.....	14
5.2 Типи інформації в схемі структури взаємодії.....	15
Додаток А (обов'язковий) Визначення схеми структури взаємодії.....	19
Додаток В (обов'язковий) Визначення шаблонів.....	75
Додаток С (довідковий) Приклад спрощеної карти взаємодії для проектного бюро..	104
Додаток D (довідковий) Принцип застосування алгоритму промотора .....	126
Бібліографія .....	128

## НАЦІОНАЛЬНИЙ ВСТУП

Цей національний стандарт ДСТУ EN ISO 29481-2:20XX (EN ISO 29481-2:2016, IDT; ISO 29481-2:2012, IDT) «Інформаційні моделі будівель. Настанова з доставляння інформації. Частина 2. Інтерактивне середовище», прийнятий методом перекладу, – ідентичний щодо EN ISO 29481-2:2016 (версія en) «Building information models — Information delivery manual — Part 2: Interaction framework» (ISO 29481-1:2012).

Технічний комітет стандартизації, відповідальний за цей стандарт в Україні, – ТК 301 «Металобудівництво».

Текст ISO 29481-2:2012 прийнято CEN як EN ISO 29481-2:2016 без будь-яких змін.

У цьому національному стандарті зазначено вимоги, які відповідають законодавству України.

До стандарту внесено такі редакційні зміни:

– слова «цей міжнародний стандарт», «ця частина стандарту» та «цей документ» замінено на «цей стандарт»;

– структурні елементи стандарту: «Титульний аркуш», «Передмову», «Національний вступ», першу сторінку, розділи «Терміни та визначення понять» та «Бібліографічні дані» – оформлено згідно з вимогами національної стандартизації України;

– у розділі 2 та «Бібліографії» наведено «Національне пояснення», виділене рамкою;

– зі «Вступу» до EN ISO 29481-2:2016 у цей «Національний вступ» внесено все, що безпосередньо стосується цього стандарту;

– рисунки наведено одразу після тексту, де вперше виконано посилання на них, або на наступній сторінці;

Копії нормативних документів, на які є посилання в цьому стандарті, можна отримати в Національному фонді нормативних документів.

## **ПЕРЕДМОВА ДО ISO 29481-2:2012**

ISO (Міжнародна організація зі стандартизації) є всесвітнім об'єднанням національних органів стандартизації (органів – членів ISO). Роботу з підготування міжнародних стандартів зазвичай виконують, залучаючи технічні комітети ISO. Кожен орган– член ISO, зацікавлений у темі, за якою створено технічний комітет, має право бути представлений у цьому комітеті. У роботі беруть участь також урядові та неурядові міжнародні організації, які взаємодіють з ISO. ISO тісно співпрацює з Міжнародною електротехнічною комісією (IEC) з усіх питань електротехнічної стандартизації.

Міжнародні стандарти розробляють відповідно до правил, викладених у директивах ISO/IEC, частина 2.

Основним завданням технічних комітетів є підготування міжнародних стандартів. Прийняті технічними комітетами проекти міжнародних стандартів розсилають до органів – членів ISO для голосування. Для публікації міжнародного стандарту потрібно схвалення щонайменше 75 % організацій – членів, які беруть участь у голосуванні.

Потрібно звернути увагу на те, що деякі елементи цього стандарту можуть бути предметом патентних прав. ISO не несе відповідальності за виявлення будь-якого чи всіх таких патентних прав.

Цей стандарт було підготовлено Технічним комітетом ISO/TC 59 «Будівлі та інженерні споруди», ПК 13 «Організація інформації щодо будівель та споруд».

Стандарт ISO 29481 під загальною назвою «Інформаційні моделі будівель. Настанова з доставляння інформації» містить такі частини:

– частина 1 «Методологія та формат»;

– частина 2 «Інтерактивне середовище».

На стадії підготування перебуває такий стандарт:

– частина 3 «Визначення проекції моделі»

## ВСТУП

Будівельне інформаційне моделювання являє собою технологічний підхід до описування та представлення інформації, застосовної до проектування, будівництва та експлуатації будівельних об'єктів. Такий підхід до управління інформацією об'єднує різні набори даних, використовуваних у сфері будівництва, в єдине інформаційне середовище, зменшуючи, а часом і усуваючи, потребу в багатьох типах традиційно використовуваної документації на паперових носіях.

Настанова з доставляння інформації (*information delivery manual; IDM*) забезпечує найбільш ефективне застосування інформаційної моделі будівлі (*building information model; BIM*). Якщо потрібна інформація є доступною, а якість цієї інформації – задовільною, це сприятиме значному поліпшенню процесу будівництва. Для цього потрібно спільне розуміння процесів життєвого циклу будівельного об'єкта та інформації, потрібної для їх реалізації та отримання результатів.

У цьому стандарті основну увагу приділено аспектам процесу будівництва, з якими пов'язано завдання з управління та координування дій залучених сторін. Координація залежить від процесу комунікації, що має бути добре структурованим, однозначно зрозумілим, докладним та оперативним. Завдяки чіткому зосередженню на питаннях координації та взаємодії цей стандарт природним чином доповнює міжнародні стандарти з будівельного інформаційного моделювання, наприклад, ISO 10303-239 та ISO 16739.

У цьому стандарті визначено методологію та формат описування заходів із координування дій між дійовими особами – учасниками будівельного проекту. У ньому описано способи виявлення та визначення процесів із координації та потрібної для їх виконання



інформації. Викладені загальні принципи взаємодії уможливають стандартизувати інтерактивне середовище для побудови процесів відповідно до потреб національного, регіонального рівня та конкретного проекту. У цьому стандарті також встановлено формат для підтримання програмних рішень, що надають постачальники розробок ІКТ. Відповідність різних програмних рішень ІКТ до вимог цього стандарту передбачає об'єднання різних систем управління процесами. Таким способом стандарт встановлює основні принципи, що забезпечують надійність обміну/спільного використання інформації користувачами для досягнення ними впевненості, що інформація, яку вони надсилають чи отримують, є точною та достатньою для координаційної діяльності, яку їм потрібно виконувати.

Розроблення цього стандарту було обумовлено потребою користувачів досягти надійності обміну інформацією. Більшість його положень засновано на нідерландському стандарті VISI (*Voorwaarden scheppen voor Invoeren Standaardisatie ICT in de Infrastructuur-sector*), розробленому в 2003 році.



# НАЦІОНАЛЬНИЙ СТАНДАРТ УКРАЇНИ

---

## ІНФОРМАЦІЙНІ МОДЕЛІ БУДІВЕЛЬ. НАСТАНОВА З ДОСТАВЛЯННЯ ІНФОРМАЦІЇ.

### ЧАСТИНА 2. ІНТЕРАКТИВНЕ СЕРЕДОВИЩЕ

BUILDING INFORMATION MODELS —  
INFORMATION DELIVERY MANUAL —  
PART 2: INTERACTION FRAMEWORK

---

Чинний від 20XX-XX-XX

## 1 СФЕРА ЗАСТОСУВАННЯ

У цьому стандарті визначено методологію та формат для описування узгоджених дій, виконуваних дійовими особами – учасниками будівельного проекту на всіх етапах життєвого циклу будівельних об'єктів.

У стандарті встановлено:

- методологію описування інтерактивного середовища;
- відповідний спосіб відображення обов'язків та взаємопов'язаних дій, якими сформовано контекст процесу управління потоками інформації;
- формат, в якому має бути зазначено структуру взаємодії.

Цей стандарт призначено для сприяння взаємодії між програмними засобами, використовуваними в процесі будівництва, переходу дійових осіб – учасників процесу будівництва до взаємодії у сфері цифрових технологій, а також забезпечення основи для обміну точною, достовірною, відтворюваною і високоякісною інформацією.

## 2 НОРМАТИВНІ ПОСИЛАННЯ

Наведені нижче нормативні документи, у повному складі чи їх частини, необхідні для застосування цього стандарту. У разі датованих посилань застосовують тільки наведені видання. У разі недатованих

прДСТУ EN ISO 29481-2:20XX

посилань потрібно користуватись останнім виданням наведених нормативних документів (разом зі змінами).

ISO 29481-1 Building information modeling – Information delivery manual – Part 1: Methodology and format

## НАЦІОНАЛЬНЕ ПОЯСНЕННЯ

ISO 29481-1 Інформаційні моделі будівель. Настанова з доставляння інформації. Частина 1. Методологія та формат

### **3 ТЕРМІНИ ТА ВИЗНАЧЕННЯ ПОНЯТЬ**

Нижче наведено терміни, вжиті в цьому стандарті, та визначення позначених ними понять.

**3.1 настанова з доставляння інформації; IDM** (*information delivery manual; IDM*)

Документація, в якій охоплено бізнес-процес та наведено детальні технічні вимоги щодо інформації, доставляння якої користувач, який виконує певну роль, має забезпечити у певний термін під час реалізації проекту.

**3.2 інтерактивне середовище; структура взаємодії** (*interaction framework*)

Формалізований опис елементів взаємодії, зокрема, визначення ролей, транзакцій, повідомлень у транзакції та елементів даних у повідомленнях.

**3.3 схема структури взаємодії** (*interaction framework schema*)

Формальний опис правил, яким має відповідати структура взаємодії.

### **3.4 схема взаємодії (*interaction schema*)**

Формальний опис правил, яким мають відповідати надіслані та отримані повідомлення.

### **3.5 промотор (*promotor*)**

Алгоритм, який створює схему взаємодії зі структури взаємодії, схеми структури взаємодії та файлу шаблонів, застосовних як вхідні дані.

### **3.6 файл шаблонів (*templates file*)**

Файл, що містить певну кількість шаблонів, незалежних від структури взаємодії, які призначено для створення схеми взаємодії.

### **3.7 VISI (*VISI*)**

Скорочення, яким позначено нідерландський стандарт, призначений для комунікації між партнерами для реалізації будівельних проектів.

**Примітка 1.** VISI означає «Voorwaarden scheppen voor Invoeren Standaardisatie ICT in de Infrastructuur-sector», що перекладають як «Створення умов для впровадження стандартизації інформаційно-комунікаційних технологій у галузі будівництва»).

## **4 ОСНОВОПОЛОЖНІ ПРИНЦИПИ**

### **4.1 Загальне положення**

Цей розділ призначено для висвітлення та докладного пояснення основних концепцій, на яких засновано цей стандарт.

### **4.2 BIM та IDM**

Будівельне інформаційне моделювання об'єднує різноманітні набори використовуваних у будівництві даних у єдине інформаційне середовище. Для забезпечення цього потрібно досягти спільного розуміння процесів будівництва та інформації, що забезпечує їх виконання та отримання результатів.

У багаточастинному стандарті ISO 29481 визначено методи розроблення настанови з доставляння інформації.

У першій частині ISO 29481 викладено методологію, застосовну до IDM, посилання на яку потрібно наводити у разі розроблення та використання відповідних IDM.

### 4.3 Компоненти IDM

Методологія та компоненти IDM описано в ISO 29481-1. У цій частині стандарту наведено ілюстрацію, на якій схематично показано, що являють собою різні компоненти IDM та як їх пов'язано між собою.

Застосування IDM можна розглядати під двома кутами зору: по-перше, як засіб упровадження вимог користувача та, по-друге, як засіб упровадження технічних рішень. У межах цих двох підходів існує певна кількість зон, за якими визначають характеристики різних компонентів IDM (див. рисунок 1).



Рисунок 1 – Зони застосування IDM

З точки зору вимог користувача, ці зони охоплюють:

– карти взаємодії, в яких описують ролі та взаємодію між ними;

- карти процесів, в яких описують загальний процес обмінювання інформацією;
- доставляння інформації, що передбачає описування потреб обміну інформацією;
- пов'язані процеси (збереження описів операцій з обмінювання інформацією);
- план-графік реалізації проекту (перелік подій у межах процесів, обумовлених контекстом проекту).

З точки зору технічних рішень, ці зони охоплюють:

- бізнес-об'єкти, для яких передбачено структурування вимог щодо обміну;
- специфікацію інформації, в якій описують основну схему процесу обмінювання інформацією;
- інформаційну модель будівлі.

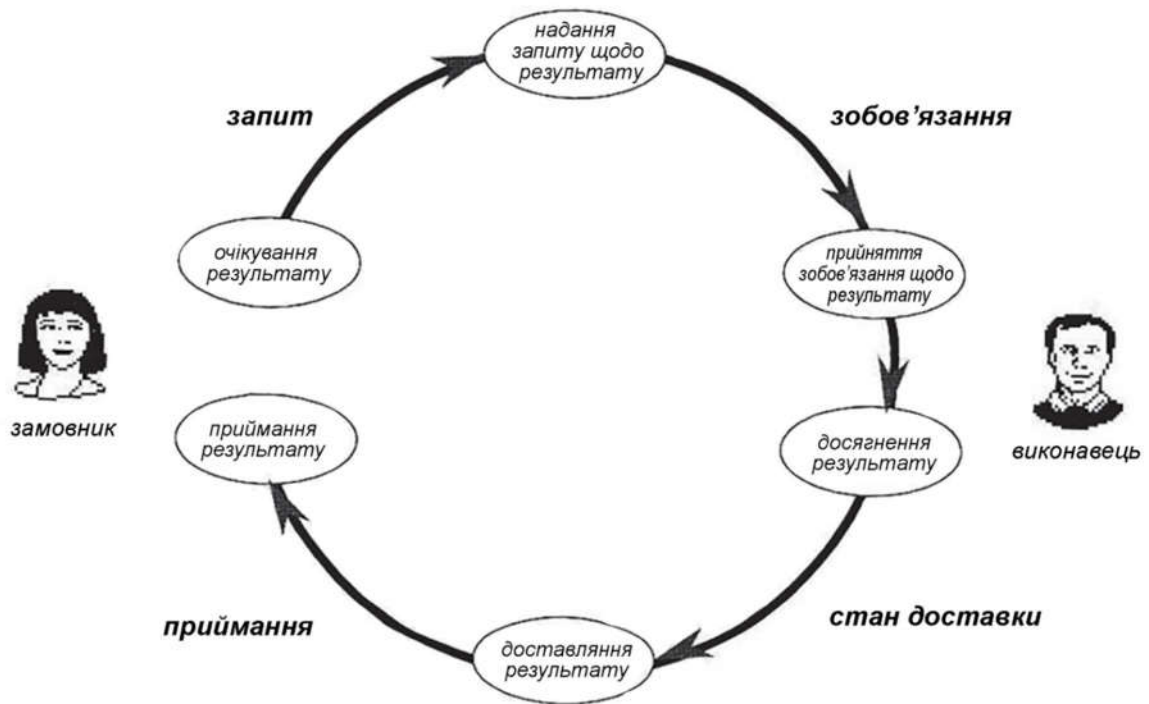
Цей стандарт установлює вимоги щодо карти взаємодії відповідно до основних принципів ділової комунікації.

#### **4.4 Основні принципи ділової комунікації**

Після отримання запиту від замовника щодо поставки товару чи надання послуги виникає ланцюжок дій, сукупний ефект яких полягає у поставлянні товару чи послуги. Такий ланцюжок дій називають бізнес-процесом. У цьому стандарті розглянуто саме первинний етап бізнес-процесу, ініційований зовнішньою стороною.

Складовою частиною бізнес-процесу є комунікація між залученими сторонами. У цьому стандарті зосереджено увагу саме на комунікації, що стосується доставляння результатів виконаної роботи (виконавча комунікація). Ініціювання та виконання запиту здійснюють за допомогою комунікативних дій. У комунікативній дії завжди беруть

участь дві сторони: особа, яка виконує дію, та особа, для якої цю дію призначено. Оброблення запиту, зазвичай, відбувається за певним шаблоном, який називають транзакцією.



**Рисунок 2** – Шаблон транзакції (Dietz J. L. G., 2006 [1])

На рисунку 2 представлено найпростішу форму такого шаблону транзакції. На ньому показано, що отримання нового результату роботи (наприклад, «очікуваного результату» – доставки документа) розпочинається з того, що дехто в ролі замовника надає декому в ролі виконавця запит щодо цього результату. Це призводить процес до фази «надання запиту щодо результату». Виконавець у відповідь на запит приймає на себе зобов'язання досягти очікуваного результату, що призводить процес до фази «прийняття зобов'язання щодо результату». Виконавець постає перед завданням: він має виконати прийняте зобов'язання, фактично підготувавши документ і прийнявши рішення про його доставку. В акті передачі документа замовнику він заявляє про виконання свого зобов'язання. Замовник реагує на цей стан прийманням, внаслідок чого результат отримано. Цим актом завершують транзакцію.

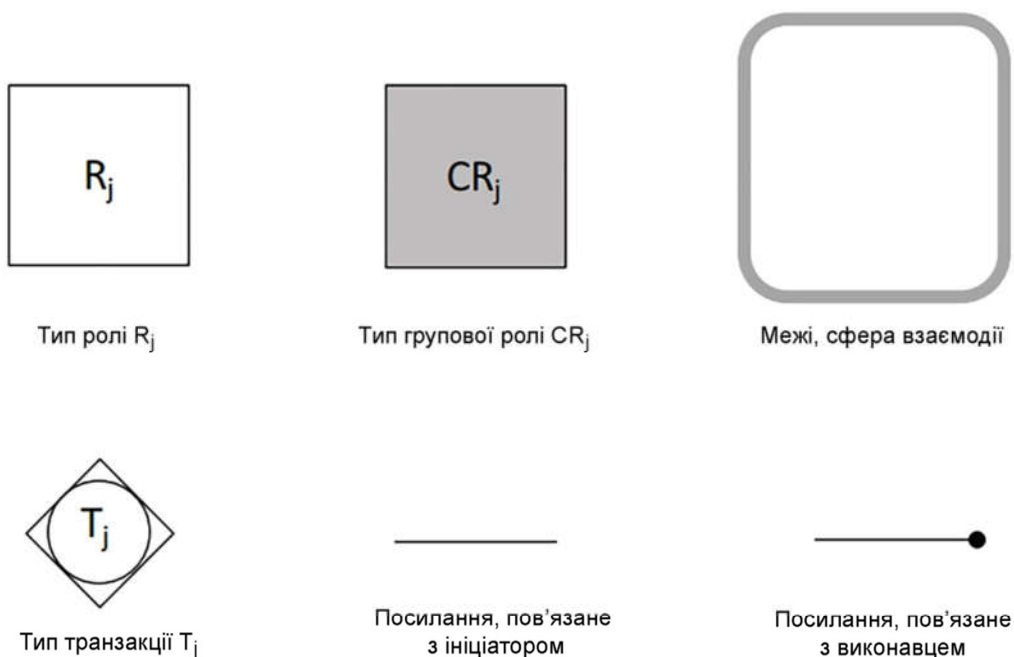


У виконанні бізнес-процесу часто беруть участь багато суб'єктів. Їх дії обумовлені їх ролями у процесі. Ролі/виконавці ролей під час підприємницької діяльності співпрацюють з іншими ролями/виконавцями ролей, здійснюючи транзакції. Відображення взаємодії між ролями/виконавцями ролей називають картою взаємодії.

#### 4.5 Карта взаємодії

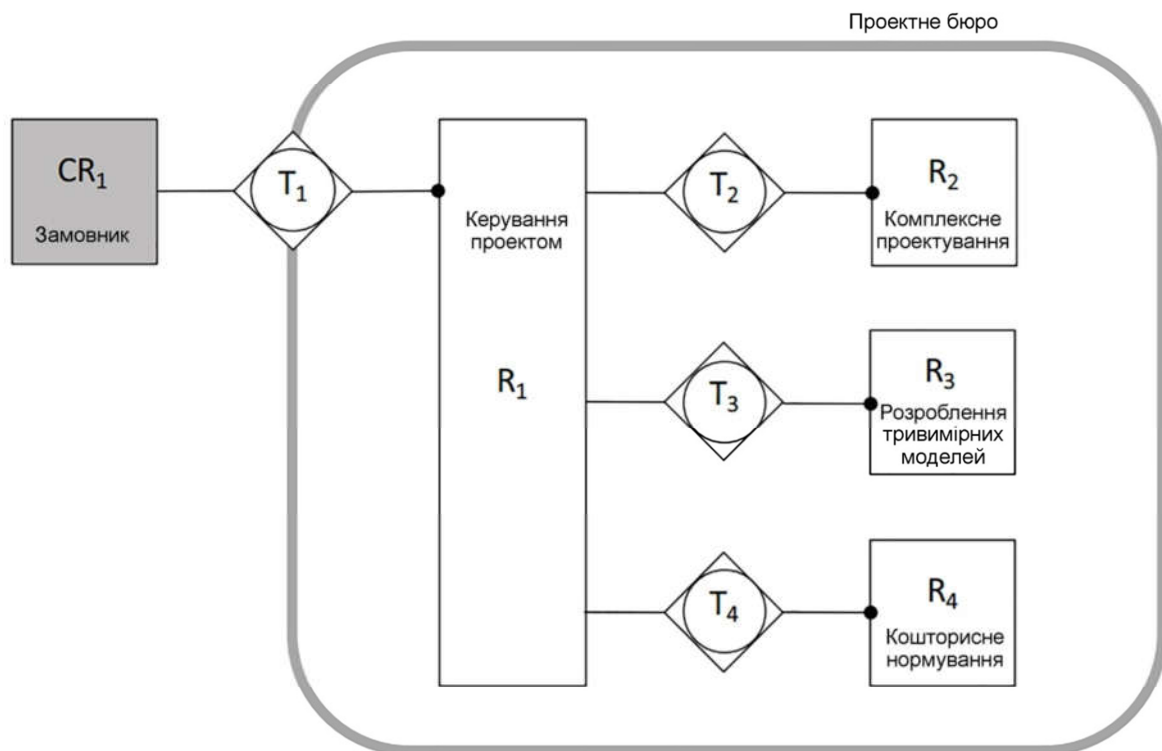
У карті взаємодії визначають відповідні типи ролей і типи транзакцій для певного процесу. В IDM розрізняють ініціувальну роль, виконавець якої надає запит, та роль виконавця, який виконує цей запит. Потрібно, щоб транзакція мала лише одну ініціувальну роль та одну виконавчу роль. На рисунку 3 зображено складові частини карти взаємодії.

**Примітка.** Нотацію, застосовну до карти взаємодії, засновано на структурі моделі, яку описано в публікації професора Яна Л. Г. Дітца. Ця нотація відрізняється від прийнятої в BPMN, і її використовують для підготування найпростіших видів карт. Крім того, до неї уведено поняття «транзакція», якого немає у BPMN.



**Рисунок 3** – Складові частини карти взаємодії

Перевага карти взаємодії полягає в тому, що в ній зосереджено увагу на взаємодії між ролями, без відображення складного процесу в інтерактивному середовищі ролей та деталей цієї взаємодії. Використання ролей в абстрактному значенні уможливорює застосовувати карту взаємодії до багатьох різних ситуацій. Карта взаємодії є цінним інструментом для аналізування та визначення основних елементів бізнес-процесу. На рисунку 4 наведено приклад спрощеної карти взаємодії для проектного бюро.



**Рисунок 4** – Приклад карти взаємодії

Карта взаємодії має містити всі транзакції, потрібні для забезпечення виконуваних робіт у BIM за всіма відповідними ролями. Всі ролі та транзакції на карті взаємодії має бути зазначено унікальними ідентифікаційними позначками та іменами. Нумерацію застосовують у довільній формі. Назву ролі визначають за основним видом діяльності виконавця ролі; це зосереджує увагу на виконуваних роботах в BIM завдяки цій ролі. Групова роль – це роль, що може мати у

своєму складі декілька ролей, але цей склад насправді невідомий чи не має значення.

Ці взаємодії можна представити у зведеній таблиці транзакцій.

**Таблиця 1** – Таблиця транзакцій до спрощеного прикладу проектного бюро

Результат транзакції	Тип транзакції
Проект доставлено	T <sub>1</sub> Доставка проекту
Специфікацію інженерної системи доставлено	T <sub>2</sub> Доставка специфікації інженерної системи
Тривимірну модель доставлено	T <sub>3</sub> Доставка тривимірної моделі
Кошторис доставлено	T <sub>4</sub> Доставка кошторису

#### 4.6 Повідомлення у транзакції

Транзакція має охоплювати набір повідомлень, якими обмінюються для досягнення певної цілі. У транзакції також зазначають ролі учасників, момент часу в межах життєвого циклу та послідовність повідомлень, які має бути надіслано (за потреби).

Прикладом транзакції є опрацювання запиту щодо розробки тривимірної моделі. На рисунку 5 показано повідомлення у транзакції у вигляді діаграми послідовності дій, у нотації UML. Транзакцію може ініціювати лише керівник проекту R1 за допомогою повідомлення «Запит щодо розробки тривимірної моделі». Інженер-розробник тривимірних моделей (роль R3) може відповісти повідомленням «Розробку виконано та направлено запит щодо схвалення». Після повідомлення «Розробку схвалено» чи «Розробку не схвалено» транзакція завершується.



**Рисунок 5** – Приклад повідомлень у транзакції

Повідомлення являє собою інформаційну модель з умістом даних. До повідомлень може бути долучено вкладені файли. Виконавча роль може отримати вимоги щодо обміну інформацією, що надсилають у вигляді вкладених файлів, а результат роботи (дані для внесення у BIM) може бути доставлено ініціувальній ролі. За допомогою транзакцій здійснюють долучення доставленої інформації до контексту процесу.

#### **4.7 Інтерактивне середовище**

Щоб розробити настанову для процесу створення та передавання інформації, потрібно описати послідовно елементи взаємодії. Такий послідовний опис називають структурою взаємодії. Межі структури взаємодії мають охоплювати:

- визначення відповідних ролей;
- транзакції;
- повідомлення у транзакції;

- послідовність повідомлень у транзакції;
- елементи даних у повідомленнях.

Структуру взаємодії може бути підготовлено для певної сфери застосування та використано як стандарт на національному та міжнародному рівнях, на рівні організації чи окремого проекту. Наприклад, у Нідерландах на національному рівні розробляють структуру інтерактивного середовища для виконання всіх контрактних процедур під час реалізації будівельного проекту. Різні організації та виконавці проектів використовують цей стандарт як шаблон, адаптуючи його до конкретних потреб.

**Приклад.** Структура взаємодії може містити атрибут *CostEstimation* як екземпляр *SimpleElementType*, який буде використано як обов'язковий елемент для певного повідомлення. Він також може містити обмеження щодо формату атрибута *CostEstimation* (наприклад, зазначення вартості лише в євро у форматі чисел із двома десятковими знаками).

## **4.8 Підтримання програмних рішень**

### **4.8.1 Загальний огляд**

Наступним етапом є забезпечення взаємодії в інтерактивному середовищі, що засновано на програмних рішеннях, для досягнення таких цілей:

- підтримання редагування структури взаємодії;
- гарантування повноти та придатності структури взаємодії;
- підтримання мобільності структури взаємодії;
- забезпечення роботи інформаційних систем;
- підтримання сумісності засобів комунікації.

Підтримання програмних рішень може бути здійснено на двох рівнях. Перший рівень стосується структури взаємодії, другий –

фактичної комунікації, яка має відповідати інтерактивному середовищу. У цьому стандарті розглянуто обидва рівні.

На рисунку 6 наведено приклад способів підтримання програмних рішень. У розділах нижче викладено пояснення.

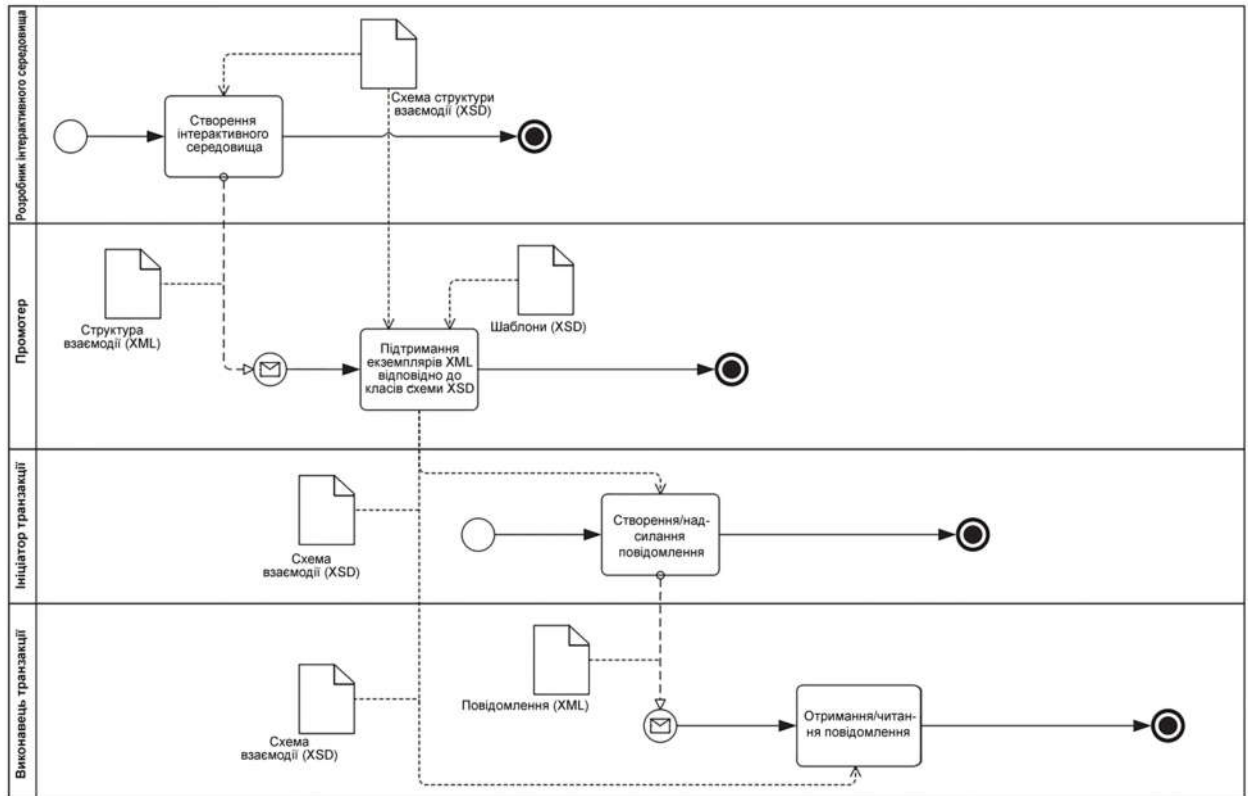


Рисунок 6 – Схема підтримання програмних рішень

#### 4.8.2 Підтримання інтерактивного середовища

Підтримання мобільності структури взаємодії потребує розуміння правил взаємодії, які має бути дотримано у межах інтерактивного середовища. Ці правила має бути охоплено схемою структури взаємодії, яку записують як файл XSD. Схему взаємодії, що містить екземпляри зазначених у ній класів, записують у вигляді файлу XML.

##### Приклад

Схема структури взаємодії установлює, які визначення атрибутів (*SimpleElementType*) та обмеження атрибутів (*UserdefinedType*) може бути уміщено до структури взаємодії.

У розділі 5 описано схему структури взаємодії та наявні класи.

Кожна структура взаємодії має бути відповідною до схеми структури взаємодії.

Редактор структури взаємодії має використовувати схему структури взаємодії для перевіряння створених структур.

### 4.8.3 Промотор

Щойно стає доступною дійсна структура взаємодії, її можливо інтерпретувати, використовуючи відповідну інформаційну систему. Опісля ця система може підтримувати процес комунікації відповідно до параметрів, заданих у структурі взаємодії. Насамкінець, бажано мати можливість перевіряти отримані та надіслані повідомлення, що можливо виконувати за допомогою схеми взаємодії.

Генерування схеми взаємодії відбувається за використання загального алгоритму, який називають «промотор». Промотор «просуває» екземпляри XML до визначених класів у схемі XSD. Використовують такі вхідні дані:

- структура взаємодії (XML);
- схема структури взаємодії (XSD);
- файл шаблонів (XSD), що містить певну кількість шаблонів, не описаних у структурі взаємодії, але дійсних для кожної схеми взаємодії, наприклад: заголовок повідомлення.

Як вихідні дані отримують схему взаємодії, записану у вигляді файлу XSD.

#### *Приклад*

Промотор отримує дані із структури взаємодії, щоб долучити атрибут *CostEstimation*, який буде використано як обов'язковий елемент для певного повідомлення, і створює схему взаємодії, яка визначає повідомлення з атрибутом *CostEstimation*.

У додатку В описано шаблони, які уміщують у файл XSD.

У додатку D викладено принципи роботи промотора.

#### **4.8.4 Підтримання комунікації**

Кожна інформаційна система, яку застосовують для комунікації відповідно до визначених правил у структурі взаємодії, має функціонувати на основі відповідної структури взаємодії та схеми взаємодії. Кожне надіслане чи отримане повідомлення має бути дійсним згідно зі схемою взаємодії.

#### **4.8.5 Технічне виконання процесу комунікації**

Технічне забезпечення процесу обмінювання повідомленнями з уміщеними файлами, застосовного до інформаційних систем, потребує розроблення настанови з його упровадження. У настанові має бути висвітлено такі теми:

- комунікаційний протокол;
- архітектура/сервери системи зв'язку;
- кодування;
- виклик функції SOAP.

Настанови з упровадження у цьому стандарті не розглянуто.

## **5 ФОРМАТ СТРУКТУРИ ВЗАЄМОДІЇ**

### **5.1 Загальні положення**

Згідно з положеннями розділу 4, для підтримання програмних рішень кожна структура взаємодії має відповідати схемі структури взаємодії. У цьому розділі визначено формат структури взаємодії за допомогою опису схеми структури взаємодії.

У підрозділі 5.2 представлено огляд можливих класів інформації у структурі взаємодії, які може бути визначено в схемі структури взаємодії. Оскільки структуру взаємодії визначають у XML, замість



класу використовують термін «тип». Додаток А містить вичерпний опис схеми взаємодії. У додатку С наведено приклад екземпляра структури взаємодії.

## **5.2 Типи інформації в схемі структури взаємодії**

### **5.2.1 Вступне положення**

Схему заповнюють великою кількістю класів або типів. У цьому розділі викладено короткий опис можливих типів для уміщення у схемі структури взаємодії. У додатку А наведено повний опис схеми взаємодії у XML. Структуру взаємодії має бути створено з екземплярів цих типів, і в її заголовку має бути зазначено типи, які визначено у схемі.

### **5.2.2 AppendixType**

*AppendixType* є визначенням, за допомогою якого визначають структуру елементів, що відносяться до метаданих. Екземпляр *AppendixType* використовують для визначення певних типів файлів чи документів, які можуть бути складовими частинами відправлених/отриманих повідомлень. Структура елементів, пов'язаних з екземпляром *AppendixType*, являє собою конкретні метадані, потрібні для певного типу файлу чи документа.

### **5.2.3 ComplexElementType**

*ComplexElementType* є сукупністю типів *SimpleElementType*. Кожен *SimpleElementType* з'являється саме стільки разів, скільки на нього посилається пов'язаний елемент.

### **5.2.4 ElementCondition**

Екземпляр *ElementCondition* описує умови застосування значень елементів у послідовності повідомлень. Наприклад, якщо екземпляр типу *ElementCondition* створюють зі значенням FIXED, це вказує на те, що елементи в послідовності повідомлень має бути скопійовано у разі

наявності одного і того самого елемента, значення якого неможливо змінити. *ElementCondition* може бути пов'язано посиланнями з різними рівнями структури. Його може бути безпосередньо пов'язано з *SimpleElement*, але *ElementCondition* можна пов'язати також із *ComplexElement* чи *MessageInTransactionType*. У цьому разі *ElementCondition* є дійсним для всіх елементів, що є частиною структури елементів/сукупності відповідних типів.

### **5.2.5 GroupType**

*GroupType* уможливорює створення кількох екземплярів групи зі своїм власним специфічним змістом. *GroupType* можна використовувати для розподілу за категоріями повідомлень у транзакції чи документів, що стосуються повідомлень у транзакції.

### **5.2.6 MessageType**

*MessageType* використовують для визначення вмісту повідомлення. Елементи, які є частиною повідомлення, своєю чергою, угруповують в один чи декілька екземплярів *ComplexElementType*.

### **5.2.7 MessageInTransactionType**

*MessageInTransactionType* (MITT) є визначенням, яке використовують для пов'язування екземплярів типу *MessageType* з екземплярами типу *TransactionType*. Тобто, один і той самий тип повідомлення може зустрічатися кілька разів у зазначеному типі транзакції і навпаки. Можна пов'язувати декілька екземплярів *MessageType* з одним екземпляром *TransactionType*, а також один екземпляр *MessageType* з декількома екземплярами *TransactionType*. Крім того, MITT уможливорює змінення напрямку дії від виконавця до ініціатора. Також він забезпечує можливість проконтролювати, чи заблоковано потік повідомлень відкритими вторинними транзакціями, чи ні.

### **5.2.8 OrganisationType**

Тип призначено для визначення певної групи організацій. Зазвичай, у структурі буває принаймні один екземпляр, появу якого обумовлено конкретною причиною, що передбачає визначення структури елементів організації.

### **5.2.9 PersonType**

Тип призначено для визначення типу особи. Зазвичай, у структурі буває принаймні один екземпляр, появу якого обумовлено конкретною причиною, що передбачає визначення структури елементів, якими визначають особу. *PersonType* можна використовувати для класифікації груп осіб, які мають відношення до якоїсь ролі.

### **5.2.10 ProjectType**

Тип призначено для визначення типу проекту. Загалом, у структурі буває принаймні один екземпляр, появу якого обумовлено конкретною причиною, що передбачає визначення структури елементів, за якими визначають проект.

### **5.2.11 RoleType**

Тип призначено для визначення ролі. Екземпляри *RoleType* є обов'язковою умовою для створення *TransactionType* у структурі.

### **5.2.12 Simple ElementType**

*SimpleElementType* є визначенням, яким описують можливі властивості в структурах об'єктів. Пов'язання з об'єктом завжди здійснюють через *ComplexElementType*.

### **5.2.13 TransactionPhaseType**

Визначення, яке може бути використано, щоб визначити екземпляр, що описує фазу, в якій перебуває транзакція. Наприклад,

екземпляр *TransactionPhaseType* може визначати фазу «запит щодо результату» або «в режимі очікування».

### 5.2.13 TransactionType

Тип призначено для визначення транзакції. В екземплярі транзакції має бути визначено ролі ініціатора та виконавця.

### 5.2.15 UserDefinedType

*UserDefinedType* використовують для встановлення типів даних (наприклад, символічного рядка) та обмежень XSD. Наприклад, за допомогою екземпляра *UserDefinedType* можна визначити мінімальну довжину рядка.

На рисунку 7 зображено модель схеми структури взаємодії, включно з усіма взаємозв'язками.

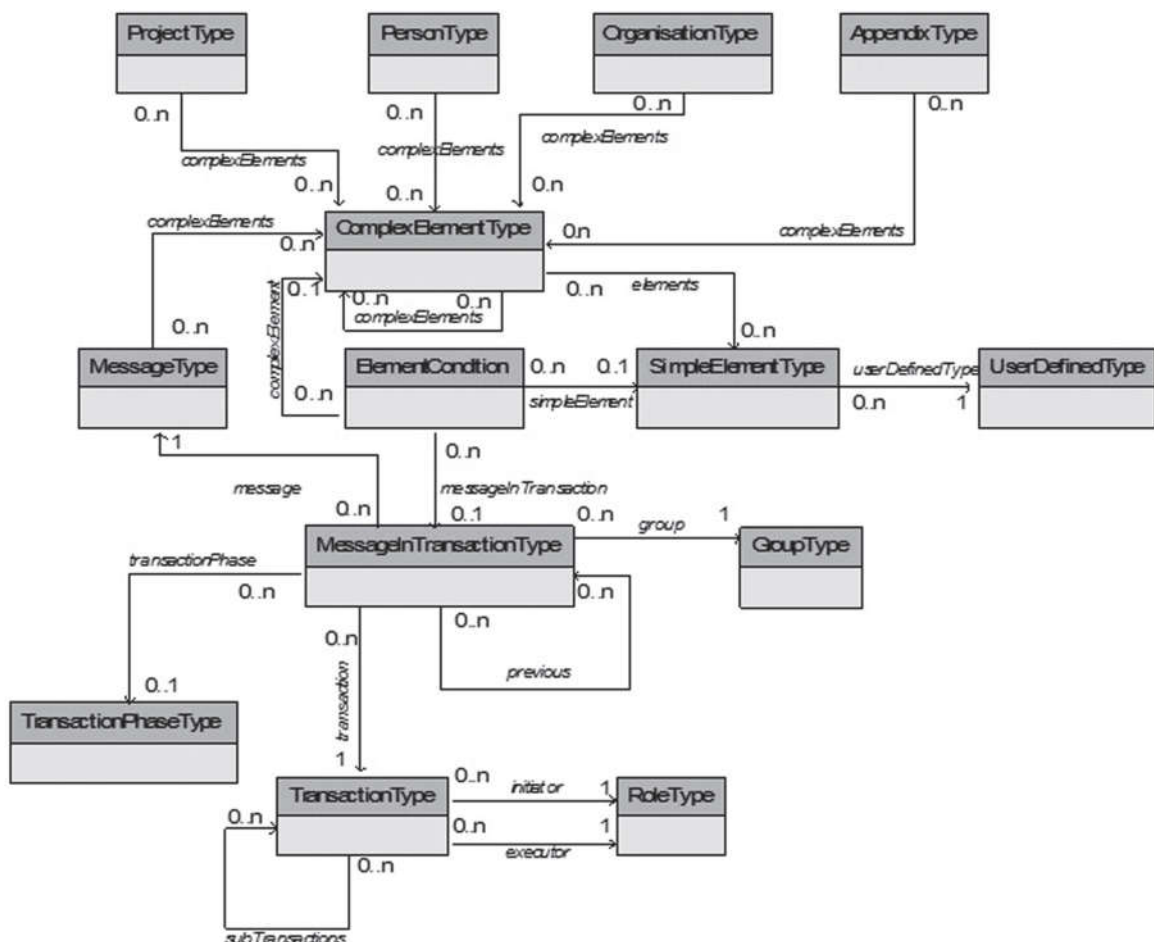


Рисунок 7 – Схема структури взаємодії із зазначенням типів та взаємозв'язків

## **ДОДАТОК А**

(обов'язковий)

### **ВИЗНАЧЕННЯ СХЕМИ СТРУКТУРИ ВЗАЄМОДІЇ**

#### **А.1 Вступні положення**

Структура взаємодії має відповідати правилам, які враховано у схемі структури взаємодії. У цьому додатку визначення схеми структури взаємодії наведено у форматі EXPRESS та в XSD. Крім того, наведено опис типів елементів, атрибутів, елементів та зв'язків.

Для всіх об'єктів у структурі взаємодії має бути визначено дати початку та завершення їх застосовності. Дотримання цієї вимоги уможливорює долучення часових обмежень щодо дієвості певного об'єкта.

#### **А.2 Визначення схеми структури взаємодії (формат EXPRESS)**

SCHEMA ISO 29481\_Part\_2A;

ENTITY ProjectType; – є визначенням конкретної групи проектів. Зазвичай, у структурі взаємодії буває присутнім лише один екземпляр, що визначає структуру елементів, які має бути представлено в кожному екземплярі проекту.

namespace: STRING;  
description: STRING;  
startDate: DATETIME;  
endDate: DATETIME;  
state: STRING;  
dateLaMu: DATETIME;  
userLaMu: STRING;  
language: OPTIONAL STRING;  
category: OPTIONAL STRING;

прДСТУ EN ISO 29481-2:20XX

helpInfo: OPTIONAL STRING;

code: OPTIONAL STRING;

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

END\_ENTITY;

ENTITY PersonType; – є визначенням конкретної групи осіб. Зазвичай, у структурі взаємодії буває присутнім лише один екземпляр, що визначає структуру елементів, які має бути представлено в кожному екземплярі особи.

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; – є визначенням конкретної групи організацій. Зазвичай, у структурі взаємодії буває присутнім лише один екземпляр, що визначає структуру елементів, які має бути представлено в кожному екземплярі організації.

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 містить визначення додатка. У розділі складеного елемента можна зазначити, які елементи даних має бути записано, використовуючи додаток.

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 містить набір SimpleElementTypes. Кожен заявлений SimpleElementType з'являється саме стільки разів, скільки на нього посилаються.

description: STRING;  
 startDate: DATETIME;  
 endDate: DATETIME;  
 state: STRING;

прДСТУ EN ISO 29481-2:20XX

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; – є визначенням типу повідомлення (MessageType), що визначає структуру повідомлення та набір типів елементів SimpleElementType (через ComplexElementType), який воно може супроводжувати.

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; – є умовою SimpleElementType у тому вигляді, як її використовують у певному MessageType.

description: STRING;

condition: STRING;



helpInfo: OPTIONAL STRING;  
 complexElement: OPTIONAL ComplexElementType;  
 simpleElement: OPTIONAL SimpleElementType;  
 messageInTransaction: OPTIONAL MessageInTransactionType;  
 END\_ENTITY;

ENTITY SimpleElementType; – є визначенням простого типу елемента (SimpleElementType). Цей тип елемента визначає властивість, яка може з'являтися в різних структурах об'єктів, наприклад, у MessageType (див. також AppendixType, ProjectType, PersonType та OrganisationType). SimpleElementType завжди є уміщеним у 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; – специфікація типу даних (для використання в SimpleElementType). Цей тип даних уміщує заповнювані проміжки в кінцевому повідомленні, наприклад, у Нідерландах поштовий індекс завжди починається з чотирьох цифр, а після них зазначають два літерні символи.

description: STRING;  
 state: STRING;

прДСТУ EN ISO 29481-2:20XX

dateLaMu: DATETIME;

userLaMu: STRING;

baseType: STRING;

xsdRestriction: OPTIONAL STRING;

language: OPTIONAL STRING;

helpInfo: OPTIONAL STRING;

END\_ENTITY;

ENTITY MessageInTransactionType; – MessageType, що з'являється у TransactionType, пов'язаний із певним типом групи (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; – є визначенням фази, пов'язаної з транзакцією. Приклади: «призначення прийнято» або «частину результату отримано».

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; – є визначенням типу транзакції. Тип транзакції може бути пов'язано посиланнями на інші типи транзакцій. Транзакцію ініціює особа, яка належить до організації, що виконує певну роль. На цьому рівні зазначають тип ініціювальної ролі (його буде застосовано у розширеній схемі). Така сама умова стосується і виконавчої ролі.

```

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;

```

прДСТУ EN ISO 29481-2:20XX

appendixTypes: OPTIONAL SET [1:?] OF AppendixType;  
END\_ENTITY;

ENTITY RoleType; – є визначенням конкретного типу ролі.

description: STRING;  
startDate: DATETIME;  
endDate: DATETIME;  
state: STRING;  
dateLaMu: DATETIME;  
userLaMu: STRING;  
language: OPTIONAL STRING;  
category: OPTIONAL STRING;  
helpInfo: OPTIONAL STRING;  
code: OPTIONAL STRING;  
responsibilityScope: OPTIONAL STRING;  
responsibilityTask: OPTIONAL STRING;  
responsibilitySupportTask: OPTIONAL STRING;  
responsibilityFeedback: OPTIONAL STRING;  
END\_ENTITY;

ENTITY GroupType; – є визначенням групи для зберігання додатків,  
надісланих із повідомленням у транзакції.

description: STRING;  
startDate: DATETIME;  
endDate: DATETIME;  
state: STRING;  
dateLaMu: DATETIME;  
userLaMu: STRING;  
language: OPTIONAL STRING;  
category: OPTIONAL STRING;  
helpInfo: OPTIONAL STRING;

END\_ENTITY;  
END\_SCHEMA;

### A.3 Визначення схеми структури взаємодії (формат XSD)

```
<?xml version="1.0" encoding = "UTF-8"? >
<xsd:schema targetNamespace="http://www.ISO 29481_Part_2A.com/schemas"
xmlns:iso29481p2a = "http://www.ISO 29481_Part_2A.com/schemas" xmlns:xsd =
"http://www.w3.org/2001/XMLSchema" elementFormDefault = "qualified" >
  <!-- root element declaration (for SCHEMA defnitions) -->
  <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>UserDefnedType"/ >
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  <!-- element declarations (for ENTITY defnitions) -->
  <xsd:element name="AppendixType" type =
"iso29481p2a:AppendixTypeType" >
    <xsd:annotation>
      <xsd:documentation>An AppendixType contains the
definition of an appendix. Which data items should be recorded with an
appendix can be specifed in the complex element section.</xsd:documentation>
```

```

        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ComplexElementType" type =
"iso29481p2a:ComplexElementTypeType" >
        <xsd:annotation>
            <xsd:documentation>A ComplexElementType contains a set
of SimpleElementTypes. Each stated SimpleElementType occurs exactly the
number of times mentioned.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ElementCondition" type =
"iso29481p2a:ElementConditionType" >
        <xsd:annotation>
            <xsd:documentation>The condition of a SimpleElementType
as used within a specifc MessageType.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="GroupType" type = "iso29481p2a:GroupTypeType" >
        <xsd:annotation>
            <xsd:documentation>The defnition of a group to store
appendices sent with a message within a transaction.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="MessageInTransactionType" type =
"iso29481p2a:MessageInTransactionTypeType" >
        <xsd:annotation>
            <xsd:documentation>The occurrence of a MessageType
within a TransactionType related to a certain group type
(GroupType).</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="MessageType" type = "iso29481p2a:MessageTypeType"
>
        <xsd:annotation>
            <xsd:documentation>The defnition of a type of message
(MessageType), specifying the structure of the message and which set of
SimpleElementType's (via ComplexElementType's) may
accompany.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="OrganisationType" type =
"iso29481p2a:OrganisationTypeType" >

```

```

        <xsd:annotation>
            <xsd:documentation>The definition of a specific group
of organisations. Generally only one instance will be present in a
interaction framework defining the structure of elements that each instance
of organisation should expose.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="PersonType" type = "iso29481p2a:PersonTypeType" >
        <xsd:annotation>
            <xsd:documentation>The definition of a specific group
of persons. Generally only one instance will be present in a interaction
framework defining the structure of elements that each instance of person
should expose.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="ProjectType" type = "iso29481p2a:ProjectTypeType"
>
        <xsd:annotation>
            <xsd:documentation>The defnition of a specifc group of
projects. Generally only one instance will be present in a interaction
framework defning the structure of elements that each instance of project
should expose.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="RoleType" type = "iso29481p2a:RoleTypeType" >
        <xsd:annotation>
            <xsd:documentation>The defnition of a specifc role
type.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="SimpleElementType" type =
"iso29481p2a:SimpleElementTypeType" >
        <xsd:annotation>
            <xsd:documentation>The defnition of a simple element
type (SimpleElementType). This element type specifes a property whic may
occur within various object structures like for example in MessageType (see
also AppendixType, ProjectType, PersonType and OrganisationType). A
SimpleElementType is always embedded in a
ComplexElementType.</xsd:documentation>
        </xsd:annotation>
    </xsd:element>
    <xsd:element name="TransactionPhaseType" type =
"iso29481p2a:TransactionPhaseTypeType" >

```

## прДСТУ EN ISO 29481-2:20XX

```
<xsd:annotation>
    <xsd:documentation>The definition of a phase related to
a transaction. Examples are 'assignment accepted' or 'result part
received'.</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TransactionType" type =
"iso29481p2a:TransactionTypeType" >
    <xsd:annotation>
        <xsd:documentation>The definition of a type of
transaction. A transaction type may reference again other transaction types.
A transaction will be initiated by a person belonging to an organisation
fulfilling a certain role. At this level the role type of the initiator should
be stated (the promoted schema will enforce this). The same holds for the
executor.</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<xsd:element name="UserDefnedType" type =
"iso29481p2a:UserDefnedTypeType" >
    <xsd:annotation>
        <xsd:documentation>A specifikation of a data type (to be
used in a SimpleElementType).</xsd:documentation>
    </xsd:annotation>
</xsd:element>
<!-- element reference declarations (for ENTITY defnitions) -->
<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:MessageInTransactionTypeTy
peRef"/ >
    <xsd:element name="MessageTypeRef" type =
"iso29481p2a:MessageTypeTypeRef"/ >
    <xsd:element name="OrganisationTypeRef" type =
"iso29481p2a:OrganisationTypeTypeRef"/ >
    <xsd:element name="PersonTypeRef" type =
"iso29481p2a:PersonTypeTypeRef"/ >
```



```

        <xsd:element name="ProjectTypeRef" type =
"iso29481p2a:ProjectTypeTypeRef" / >
        <xsd:element name="RoleTypeRef" type = "iso29481p2a:RoleTypeTypeRef" /
>
        <xsd:element name="SimpleElementTypeRef" type =
"iso29481p2a:SimpleElementTypeTypeRef" / >
        <xsd:element name="TransactionPhaseTypeRef" type =
"iso29481p2a:TransactionPhaseTypeTypeRef" / >
        <xsd:element name="TransactionTypeRef" type =
"iso29481p2a:TransactionTypeTypeRef" / >
        <xsd:element name="UserDefnedTypeRef" type =
"iso29481p2a>UserDefnedTypeTypeRef" / >
        <!-- complex types for element declarations (for ENTITY definitions) -->
        <xsd:complexType name="AppendixTypeType" >
            <xsd:complexContent>
                <xsd:extension base="iso29481p2a:elementType" >
                    <xsd:sequence>
                        <xsd:element name="description" type =
"xsd:string" / >
                        <xsd:element name="startDate" type =
"xsd:dateTime" / >
                        <xsd:element name="endDate" type =
"xsd:dateTime" / >
                        <xsd:element name="state" type =
"xsd:string" / >
                        <xsd:element name="dateLaMu" type =
"xsd:dateTime" / >
                        <xsd:element name="userLaMu" type =
"xsd:string" / >
                        <xsd:element name="language" type =
"xsd:string" minOccurs = "0" / >
                        <xsd:element name="category" type =
"xsd:string" minOccurs = "0" / >
                        <xsd:element name="helpInfo" type =
"xsd:string" minOccurs = "0" / >
                        <xsd:element name="code" type =
"xsd:string" minOccurs = "0" / >
                        <xsd:element name="complexElements"
minOccurs = "0" >
                            <xsd:complexType>
                                <xsd:choice
maxOccurs="unbounded" >

```

```

                                                                 <xsd:element
ref="iso29481p2a:ComplexElementType"/ >
                                                                 <xsd:element
ref="iso29481p2a:ComplexElementTypeRef"/ >
                                                                 </xsd:choice>
                                                                 </xsd:complexType>
                                                                 </xsd:element>
                                                                 </xsd:sequence>
                                                                 </xsd:extension>
                                                                 </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ComplexElementTypeType" >
  <xsd:complexContent>
    <xsd:extension base="iso29481p2a:elementType" >
      <xsd:sequence>
        <xsd:element name="description" type =
"xsd:string"/ >
        <xsd:element name="startDate" type =
"xsd:dateTime"/ >
        <xsd:element name="endDate" type =
"xsd:dateTime"/ >
        <xsd:element name="state" type =
"xsd:string"/ >
        <xsd:element name="dateLaMu" type =
"xsd:dateTime"/ >
        <xsd:element name="userLaMu" type =
"xsd:string"/ >
        <xsd:element name="language" type =
"xsd:string" minOccurs = "0"/ >
        <xsd:element name="category" type =
"xsd:string" minOccurs = "0"/ >
        <xsd:element name="helpInfo" type =
"xsd:string" minOccurs = "0"/ >
        <xsd:element name="complexElements"
minOccurs = "0" >
          <xsd:complexType>
            <xsd:choice
maxOccurs="unbounded" >
              <xsd:element
ref="iso29481p2a:ComplexElementType"/ >
              <xsd:element
ref="iso29481p2a:ComplexElementTypeRef"/ >

```

```

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

```

```

                                <xsd:choice>
                                    <xsd:element
ref="iso29481p2a:SimpleElementType"/ >
                                    <xsd:element
ref="iso29481p2a:SimpleElementTypeRef"/ >
                                </xsd:choice>
                            </xsd:complexType>
                    </xsd:element>
                <xsd:element
name="messageInTransaction" minOccurs = "0" >
                    <xsd:complexType>
                        <xsd:choice>
                            <xsd:element
ref="iso29481p2a:MessageInTransactionType"/ >
                            <xsd:element
ref="iso29481p2a:MessageInTransactionTypeRef"/ >
                        </xsd:choice>
                    </xsd:complexType>
                </xsd:element>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>

</xsd:complexType>
<xsd:complexType name="GroupTypeType" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementType" >
            <xsd:sequence>
                <xsd:element name="description" type =
"xsd:string"/ >
                <xsd:element name="startDate" type =
"xsd:dateTime"/ >
                <xsd:element name="endDate" type =
"xsd:dateTime"/ >
                <xsd:element name="state" type =
"xsd:string"/ >
                <xsd:element name="dateLaMu" type =
"xsd:dateTime"/ >
                <xsd:element name="userLaMu" type =
"xsd:string"/ >
                <xsd:element name="language" type =
"xsd:string" minOccurs = "0"/ >
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

                                <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="frstMessage" 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>

```

```

minOccurs = "0" >
    <xsd:element name="appendixTypes"
        <xsd:complexType>
            <xsd:choice
                <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="MessageType" >
    <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>

```

```

maxOccurs="unbounded" >
                                <xsd:choice
                                <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:string" minOccurs = "0"/ >
"xsd:string" minOccurs = "0"/ >
"xsd:string" minOccurs = "0"/ >
"xsd:string" minOccurs = "0"/ >
minOccurs = "0" >
maxOccurs="unbounded" >
ref="iso29481p2a:ComplexElementType"/ >
ref="iso29481p2a:ComplexElementTypeRef"/ >
<xsd:complexType>
  <xsd:choice>
    <xsd:element
    <xsd:element
  </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"/ >

```

## прДСТУ EN ISO 29481-2:20XX

```
"xsd:string" minOccurs
"xsd:string" minOccurs = "0" / >
"xsd:string" minOccurs = "0" / >
minOccurs = "0" >
maxOccurs="unbounded" >
ref="iso29481p2a:ComplexElementType" / >
ref="iso29481p2a:ComplexElementTypeRef" / >
<xsd:element name="category" type =
= "0" / >
<xsd:element name="helpInfo" type =
<xsd:element name="code" type =
<xsd:element name="complexElements"
<xsd:complexType>
<xsd:choice
<xsd:element
<xsd:element
</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:string" minOccurs = "0" / >
"xsd:string" minOccurs = "0" / >
"xsd:string" minOccurs = "0" / >
"xsd:string" minOccurs = "0" / >
minOccurs = "0" >
maxOccurs="unbounded" >
ref="iso29481p2a:ComplexElementType" / >
ref="iso29481p2a:ComplexElementTypeRef" / >
<xsd:complexType>
  <xsd:choice>
    <xsd:element
    <xsd:element
    </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" / >

```

## прДСТУ EN ISO 29481-2:20XX

```
<xsd:element name="category" type =
"xsd:string" minOccurs = "0"/ >
<xsd:element name="helpInfo" type =
"xsd:string" minOccurs = "0"/ >
<xsd:element name="code" type =
"xsd:string" minOccurs = "0"/ >
<xsd:element
name="responsibilityScope" type = "xsd:string" minOccurs = "0"/ >
<xsd:element name="responsibilityTask"
type = "xsd:string" minOccurs = "0"/ >
<xsd:element
name="responsibilitySupportTask" type = "xsd:string" minOccurs = "0"/ >
<xsd:element
name="responsibilityFeedback" type = "xsd:string" minOccurs = "0"/ >
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SimpleElementType" >
<xsd:complexContent>
<xsd:extension base="iso29481p2a:elementType" >
<xsd:sequence>
<xsd:element name="description" type =
"xsd:string"/ >
<xsd:element name="interfaceType" type
= "xsd:string"/ >
<xsd:element name="state" type =
"xsd:string"/ >
<xsd:element name="dateLaMu" type =
"xsd:dateTime"/ >
<xsd:element name="userLaMu" type =
"xsd:string"/ >
<xsd:element name="language" type =
"xsd:string" minOccurs = "0"/ >
<xsd:element name="category" type =
"xsd:string" minOccurs = "0"/ >
<xsd:element name="helpInfo" type =
"xsd:string" minOccurs = "0"/ >
<xsd:element name="valueList" type =
"xsd:string" minOccurs = "0"/ >
<xsd:element name="userDefinedType" >
<xsd:complexType>
```

```

                                <xsd:choice>
                                    <xsd:element
ref="iso29481p2a:UserDefinedType"/ >
                                    <xsd:element
ref="iso29481p2a:UserDefinedTypeRef"/ >
                                </xsd:choice>
                            </xsd:complexType>
                        </xsd:element>
                    </xsd:sequence>
                </xsd:extension>
            </xsd:complexContent>
        </xsd:complexType>
        <xsd:complexType name="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:elementType" >

```

```

<xsd:sequence>
    <xsd:element name="description" type =
"xsd:string"/ >
    <xsd:element name="startDate" type =
"xsd:dateTime"/ >
    <xsd:element name="endDate" type =
"xsd:dateTime"/ >
    <xsd:element name="state" type =
"xsd:string"/ >
    <xsd:element name="dateLaMu" type =
"xsd:dateTime"/ >
    <xsd:element name="userLaMu" type =
"xsd:string"/ >
    <xsd:element name="language" type =
"xsd:string" minOccurs = "0"/ >
    <xsd:element name="category" type =
"xsd:string" minOccurs = "0"/ >
    <xsd:element name="helpInfo" type =
"xsd:string" minOccurs = "0"/ >
    <xsd:element name="code" type =
"xsd:string" minOccurs = "0"/ >
    <xsd:element name="result" type =
"xsd:string" minOccurs = "0"/ >
    <xsd:element name="subTransactions"
minOccurs = "0" >
        <xsd:complexType>
            <xsd:choice
maxOccurs="unbounded" >
                <xsd:element
ref="iso29481p2a:TransactionType"/ >
                <xsd:element
ref="iso29481p2a:TransactionTypeRef"/ >
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name="initiator" >
        <xsd:complexType>
            <xsd:choice>
                <xsd:element
ref="iso29481p2a:RoleType"/ >
                <xsd:element
ref="iso29481p2a:RoleTypeRef"/ >
            </xsd:choice>
        </xsd:complexType>
    </xsd:element>

```

```

        </xsd:choice>
    </xsd:complexType>
</xsd:element>
<xsd:element name="executor" >
    <xsd:complexType>
        <xsd:choice>
            <xsd:element
ref="iso29481p2a:RoleType"/ >
            <xsd:element
ref="iso29481p2a:RoleTypeRef"/ >
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
<xsd:element name="appendixTypes"
minOccurs = "0" >
    <xsd:complexType>
        <xsd:choice
maxOccurs="unbounded" >
            <xsd:element
ref="iso29481p2a:AppendixType"/ >
            <xsd:element
ref="iso29481p2a:AppendixTypeRef"/ >
        </xsd:choice>
    </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="UserDefinedTypeType" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementType" >
            <xsd:sequence>
                <xsd:element name="description" type =
"xsd:string"/ >
                <xsd:element name="state" type =
"xsd:string"/ >
                <xsd:element name="dateLaMu" type =
"xsd:dateTime"/ >
                <xsd:element name="userLaMu" type =
"xsd:string"/ >
                <xsd:element name="baseType" type =
"xsd:string"/ >
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

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

```



```

<xsd:complexType name="OrganisationTypeTypeRef" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementTypeRef"/ >
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonTypeTypeRef" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementTypeRef"/ >
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ProjectTypeTypeRef" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementTypeRef"/ >
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="RoleTypeTypeRef" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementTypeRef"/ >
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SimpleElementTypeTypeRef" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementTypeRef"/ >
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TransactionPhaseTypeTypeRef" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementTypeRef"/ >
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="TransactionTypeTypeRef" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementTypeRef"/ >
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="UserDefinedTypeTypeRef" >
    <xsd:complexContent>
        <xsd:extension base="iso29481p2a:elementTypeRef"/ >
    </xsd:complexContent>
</xsd:complexType>
<!-- group declarations (for SELECT data type definitions) -->
<!-- enumeration type declarations (for ENUMERATION data type
definitions) -->

```

## прДСТУ EN ISO 29481-2:20XX

```
<!-- simple type declarations (for TYPE defined data type definitions)
→
<!-- standard declarations, elementType, simpleType, logical (for each
translation) →
<xsd:complexType name="elementType" >
    <xsd:attribute name="id" type = "xsd:ID" use = "required"/ >
</xsd:complexType>
<xsd:complexType name="elementTypeRef" >
    <xsd:attribute name="idref" type = "xsd:IDREF" use =
"required"/ >
</xsd:complexType>
<xsd:simpleType name="logical" >
    <xsd:restriction base="xsd:string" >
        <xsd:enumeration value="true"/ >
        <xsd:enumeration value="false"/ >
        <xsd:enumeration value="unknown"/ >
    </xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

### **A.4 Типи елементів**

#### **A.4.1 AppendixType**

**Атрибути:** id

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

**Взаємозв'язок із:** *complexType*

Тип *AppendixType* містить визначення додатка. У розділі складеного елемента можна зазначити, які елементи даних має бути записано за використання додатка.

#### **Приклад**

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

```
<userLaMu>bapa</userLaMu>
</AppendixType>
```

**Примітка.** Пов'язані елементи `appendixTypes`. Якщо у структурі взаємодії визначено багато типів додатків, користувачам у певних ситуаціях буває складно обрати відповідний тип додатка. Цей взаємозв'язок уможливорює обирати з набору всіх типів додатків саме ті, що є дійсними для заданої транзакції чи повідомлення.

## A.4.2 ComplexElementType

**Атрибути:** `id`

**Елементи:** `description`, `startDate`, `endDate`, `state`, `dateLaMu`, `userLaMu`, `language`, `category`, `helpinfo`

**Взаємозв'язок із:** `elements`

Тип `ComplexElementType` містить набір типів `SimpleElementTypes`. Кожен зазначений `SimpleElementType` з'являється саме стільки разів, скільки на нього посилаються.

### Приклад

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

## A.4.3 ElementCondition

**Атрибути:** `id`

**Елементи:** `description`, `requiredNotify`, `minValue`, `maxValue`, `format`, `helpinfo`

### **Посилання: *message, element***

Умова *SimpleElementType*, яку використовують у певному *MessageType*.

#### **Приклад**

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

### **A.4.4 GroupType**

**Атрибути:** *id*

**Елементи:** *description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpinfo*

Тип є визначенням групи для зберігання додатків, надісланих із повідомленням у транзакції.

#### **Приклад**

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

### **A.4.5 MessageType**

**Атрибути:** *id*

**Елементи:** *description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpinfo, code*

**Посилання:** *complexType, appendixTypes*

Є визначенням типу повідомлення (*MessageType*), що зазначає структуру повідомлення та набір типів елементів *SimpleElementType* (через *ComplexElementType*), який він може супроводжувати.

#### Приклад

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

**Примітка.** Є елементом *firstMessage*. Зазвичай, транзакцію розпочинає особа, яка виконує роль ініціатора, за допомогою типу повідомлення, що не залежить від попереднього повідомлення жодного типу. Однак цей принцип не можна застосовувати до транзакції, яка відбувається як підпорядкована щодо іншої транзакції. Цей елемент чітко вказує, чи можна використовувати певний тип повідомлення, щоб розпочати нову транзакцію.

### A.4.6 MessageInTransactionType

**Атрибути:** *id*

**Елементи:** *requiredNotify, dateLaMu, userLaMu, received, send, state, initiatorToExecutor, openSecondaryTransactionsAllowed, firstMessage*

**Посилання:** *message, previous, transaction, transactionPhase, group, appendixTypes*

Появу типу *MessageType* у *TransactionType*, пов'язують посиланням із певним типом групи (*GroupType*).

#### *Приклад*

```
<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**

**Атрибути:** id

**Елементи:** *description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpinfo, code*

**Взаємозв'язок із:** *complexElements*

Є визначенням конкретної групи організацій. Зазвичай, у структурі взаємодії буває лише один екземпляр, що визначає структуру елементів, які має бути представлено в кожному екземплярі організації.

*Приклад*

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

#### **A.4.8 PersonType**

**Атрибути:** id

**Елементи:** *description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpinfo, code*

**Взаємозв'язок із:** *complexElements*

Є визначенням конкретної групи осіб. Зазвичай, у структурі взаємодії буває лише один екземпляр, що визначає структуру елементів, які має бути представлено в кожному екземплярі особи.

*Приклад*

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

#### **A.4.9 ProjectType**

**Атрибути:** id

**Елементи:** *namespace, description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpinfo, code*

**Взаємозв'язок із:** *complexElements*

Є визначенням конкретної групи проектів. Зазвичай, у структурі взаємодії буває лише один екземпляр, що визначає структуру елементів, які має бути представлено в кожному екземплярі проекту.

##### *Приклад*

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

#### **A.4.10 RoleType**

**Атрибути:** id

**Елементи:** *description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpinfo, code, responsibilityFeedback, responsibilityScope, responsibilitySupportTask, responsibilityTask*

Є визначенням конкретного типу ролі.

##### *Приклад*

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



```

<language/>
<category/>
<helpInfo/>
<code/>
<responsibilityScope/>
<responsibilityTask/>
<responsibilitySupportTask/>
<responsibilityFeedback/>
</RoleType>

```

#### A.4.11 SimpleElementType

**Атрибути:** id

**Елементи:** *description, interfaceType, state, dateLaMu, userLaMu, language, category, helpinfo, valueList*

**Посилання:** *composedOf, userDefinedType*

Є визначенням простого типу елемента (*SimpleElementType*). Цей тип елемента зазначає властивість, яка може виникати в різних структурах об'єктів, наприклад у *MessageType* (див. також *AppendixType, ProjectType, PersonType* та *OrganisationType*). *SimpleElementType* завжди уміщують у *ComplexElementType*.

##### Приклад

```

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

```

#### A.4.12 TransactionPhaseType

**Атрибути:** id

**Елементи:** *description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpinfo, code*

Є визначенням фази транзакції. Приклади: «призначення прийнято» або «отримано частину результату».

#### *Приклад*

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

### **A.4.13 TransactionType**

**Атрибути:** id

**Елементи:** *description, startDate, endDate, state, dateLaMu, userLaMu, language, category, helpinfo, code, result*

**Посилання:** *initiator, executor, subTransactions*

Є визначенням типу транзакції. Тип транзакції може бути пов'язано посиланнями на інші типи транзакцій. Транзакцію ініціює особа, яка належить до організації, що виконує певну роль. На цьому рівні зазначають тип ініціювальної ролі (його буде застосовано у розширеній схемі). Така сама умова стосується і виконавчої ролі.

#### *Приклад*

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

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

#### **A.4.14 UserDefinedType**

**Атрибути:** #id

**Елементи:** *description, state, dateLaMu, userLaMu, baseType, xsdRestriction, language, helpinfo*

Специфікація типу даних (для використання в *SimpleElementType*). Цей тип даних уміщує заповнювані проміжки в кінцевому повідомленні.

##### *Приклад*

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

### **A.5 Атрибути**

#### **A.5.1 id**

Унікальне «коротке» ім'я події. Після просування у структурі це ім'я стає ім'ям об'єкта.

##### *Приклад*

```
Framework object
<OrganisationType id="TNO Building and Construction" >
  <description>The attributes of an organisation</description>
  <startDate>2007-12-12T00:00:00Z</startDate>
  <endDate>9999-12-31T00:00:00Z</endDate>
  <state>active</state>
  <dateLaMu>2007-12-12T00:00:00Z</dateLaMu>
  <userLaMu>testmanagement</userLaMu>
  <language>NL</language>
  <category>S</category>
  <helpInfo>http://.../</helpInfo>
  <code>DEFAULT</code>
```

```
</OrganisationType>
Message object
  <name>TNO Building & Construction</name>
  <state>active</state>
  <dateLaMu>2007-12-02T00:00:00Z</dateLaMu>
  <userLaMu>Peter Bonsma</userLaMu>
</Organisation>
```

## A.6 Елементи

### A.6.1 baseType

Містить базовий тип *SimpleElementType*.

#### Приклад

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

У цьому прикладі *SimpleElementType* зі значенням «Height» завжди містить ціле число (можливо, з обмеженням: *xsdRestriction*).

### A.6.2 category

Категорія, пов'язана із заданим об'єктом (необов'язкове значення).

### A.6.3 code

Узгоджений код структури взаємодії для зазначеного об'єкта.

#### Приклад

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

#### A.6.4 dateLaMu

Дата та час останньої зміни зазначеного об'єкта.

*Приклад*

```

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

```

#### A.6.5 description

Опис об'єкта.

*Приклад*

```

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

```

#### A.6.6 endDate

Дата та час закінчення терміну застосовності зазначеного об'єкта.

*Приклад*

```

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

```

#### A.6.7 firstMessage

Необов'язкове логічне значення, яким зазначають, що цей об'єкт *MessageInTransactionType* може бути використано як початкове повідомлення для нової транзакції. Значення за замовчуванням – *false* (помилковий).

*Приклад*

```

<MessageInTransactionType id="..." >
  ...

```

прДСТУ EN ISO 29481-2:20XX

```
<firstMessage>>true</firstMessage>
...
</MessageInTransactionType>
```

### **A.6.8 format**

Схема компонування елемента (необов'язковий).

### **A.6.9 helpinfo**

URL-адреса або URI-код посилання на додаткову інформацію про заданий об'єкт.

#### *Приклад*

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

### **A.6.10 initiatorToExecutor**

Логічне значення, яким представлено напрямок, в якому має бути надіслано повідомлення.

#### *Приклад*

```
<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="..." >

```

Передбачено надсилання повідомлення *TenderAcceptance* від замовника (у ролі виконавця) до дійової особи (у ролі ініціатора).

### A.6.11 interfaceType

Інтерфейс типу або представлення цього *SimpleElementType* для певного повідомлення.

### A.6.12 language

Мова, яку буде використано після просування цього об'єкта у структурі.

#### Приклад

```

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

```

### A.6.13 namespace

Ім'я цільового простору імен для ідентифікації повідомлень, що належать до цієї структури взаємодії.

#### Приклад

```

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

```

**Примітка.** Елемент *namespace*. Раніше всі схеми структури взаємодії було пов'язано посиланням на один і той самий цільовий простір імен, який іноді

збігався із простором імен самої схеми взаємодії. Використовуючи цей елемент, для кожної схеми взаємодії можна визначати її власний простір імен.

#### **A.6.14 openSecondaryTransactionsAllowed**

Необов'язкове логічне значення, яке уможливорює продовження основної транзакції, навіть якщо не всі підпорядковані транзакції завершено. Значення «TRUE» інтерпретують як дозвільний сигнал для продовження (шлюз OR). Значення «FALSE» інтерпретують як заборонний сигнал для продовження допоки не завершено всі підпорядковані транзакції (шлюз AND). Якщо *openSecondaryTransactionsAllowed* не зазначено, цю умову інтерпретують як «TRUE».

#### **A.6.15 received**

Логічне значення, яке вказує на те, що попереднє повідомлення отримано.

#### **A.6.16 requiredNotify**

Із цим елементом не пов'язано жодного семантичного значення.

#### **A.6.17 responsibilityFeedback**

Очікуваний зворотний зв'язок у напрямку до інших ролей (необов'язковий рядок).

#### **A.6.18 responsibilityScope**

Сфера дії/межі визначених обов'язків для ролі (необов'язковий рядок).

#### **A.6.19 responsibilitySupportTask**

Завдання, які має бути виконано, щоб забезпечити виконання інших ролей. Наприклад, делеговані обов'язки (необов'язковий рядок).



**A.6.20 responsibilityTask**

Завдання, що обумовлені обов'язками, зазначеними для цієї ролі (необов'язковий рядок).

**A.6.21 result**

Очікуваний результат від виконання транзакції.

**A.6.22 send**

Логічне значення, що вказує, надіслано повідомлення чи ні.

**A.6.23 startDate**

Дата та час початку застосовності зазначеного об'єкта.

*Приклад*

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

**A.6.24 state**

Стан видимості зазначеного об'єкта, можливі значення:

*Приклад*

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

та

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

**A.6.25 userLaMu**

Користувач останньої модифікації зазначеного об'єкта.

*Приклад*

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

### **A.6.26 valueList**

Список значень, відокремлених крапкою з комою, які може бути сприйнято зразком події на рівні повідомлення. Раніше цей елемент було призначено для підтримання переліків. Наразі цю функцію реалізують за допомогою типу елемента *UserDefinedType* та елемента *xsdRestriction*. У *xsdRestriction* зазначають значення переліку. Жодного іншого семантичного значення з цим елементом не пов'язано.

*Приклад*

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

### **A.6.27 xsdRestriction**

Цей елемент визначає обмеження, яке має бути виконано на рівні повідомлення у *SimpleElementType* цього *UserDefinedType*.

## **A.7 Посилання**

### **A.7.1 appendixTypes**

Набір типів додатків, які можна обрати.

### **A.7.2 complexElements**

Посилання на набір *SimpleElementType* (у сукупності *ComplexElementType*).

*Приклад*

```
<... 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 рівні повідомлення це детально описують так:**

```

<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>

```

```

</elementsSet2>
<elementsSet3>
  <ElementsSet3>
    ...
  </ElementsSet3>
  ...
  <ElementsSet3>
    ...
  </ElementsSet3>
</elementsSet3>
</...>

```

### A.7.3 composedOf

*SimpleElementType* може бути складено з набору *SimpleElementType* (у сукупності *ComplexElementType*).

#### Приклад

```

<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>

```

```
</composedOf>
</SimpleElementType>
```

На рівні повідомлення це описують детально:

```
<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

Умова для типу *SimpleElementType*, який буде використано в *MessageType*.

#### Приклад

```
<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>

У прикладі зазначено, що в типі повідомлення *MessageType*, позначеному як «М», елемент *Doorheight* (висота дверей) має становити щонайменше 2000, хоча зазвичай таку умову не застосовують на рівні *SimpleElementType*.

### A.7.5 elements

Набір *SimpleElementType*, доступний у межах типу *ComplexElementType*.

#### Приклад

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

На рівні повідомлення це описують детально:

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

</...>

Обов'язковою умовою є те, що всі елементи точно зазначають лише один раз. У цих дверей завжди має бути верхнє вікно. Кількість дверей не обмежено.

### **A.7.6 executor**

«Виконавча» роль (*RoleType*), пов'язана з певною транзакцією.

#### *Приклад*

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

### **A.7.7 group**

*GroupType*, пов'язаний із повідомленням у конкретній транзакції.

#### *Приклад*

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

У прикладі повідомлення «М» у транзакції «Т» належить до групи «G» (у транзакції «Т» може бути більше повідомлень «М», які належать до тієї самої чи до іншої групи).

### A.7.8 initiator

Роль «ініціювання» (*RoleType*), яка належить до певної транзакції.

#### Приклад

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

### A.7.9 message

Повідомлення в *MessageInTransactionType*.

#### Приклад

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

### A.7.10 previous

*MessageInTransactionType* може забезпечити виконання попереднього повідомлення в межах певної транзакції (потрібно, щоб цей тип попереднього повідомлення *MessageInTransactionType* не належав до одного і того самого *TransactionType*).



**Приклад**

```

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

```

Транзакції, які можливо виконувати в межах цієї транзакції.

**Приклад**

```

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

```

Тип транзакції *TransactionType*, зазначений як «*AcquireWork*» містить у своєму складі такі типи *TransactionType*, як *AcquisitionTrack* та *TenderTraject*.

## A.7.11 simpleElements

Набір простих типів елементів, що належать до зазначеного складеного типу елементів.

### Приклад

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

На рівні повідомлення це може призвести до такого результату:

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

## A.7.12 subTransactions

Транзакції, запуск яких можливий із заданої транзакції.

*Приклад*

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

**A.7.13 transaction**

Транзакція в межах *MessageInTransactionType*.

*Приклад*

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

**A.7.14 transactionPhase**

Елемент *TransactionPhase* для певного типу *MessageType* у конкретному типі *TransactionType*.

*Приклад*

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

```

    </TransactionType>
</transaction>
...
<transactionPhase>
    <TransactionPhaseType id="TP" >
        ...
    </TransactionPhaseType>
</transactionPhase>
...
</MessageInTransactionType>

```

У прикладі тип *MessageType*, зазначений як «М» у межах конкретного типу *TransactionType*, зазначеного як «Т», визначає *TransactionPhaseType*, зазначений як «ТР».

### A.7.15 userDefinedType

Посилання на *UserDefinedType*, за яким визначають формат типу *SimpleElementType*.

#### Приклад

```

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

```

У прикладі тип *SimpleElementType*, зазначений як «Height», визначає формат одиниці виміру для кожного зразка події (можливо, з обмеженням як *xsdRestriction*).

## ДОДАТОК В

(обов'язковий)

### ВИЗНАЧЕННЯ ШАБЛОНІВ

#### В.1 Вступне положення

Після того, як визначено застосовну структуру взаємодії, буває потрібним файл схеми взаємодії, в якому розміщено правила для фактичної комунікації. Правила, які не обумовлюють фактичну комунікацію, розміщують у файл шаблонів. У цьому додатку наведено визначення шаблонів у форматі EXPRESS. Також викладено описи типів елементів, атрибутів, елементів та посилань.

#### В.2 Визначення шаблонів

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;

прДСТУ EN ISO 29481-2:20XX  
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;

прДСТУ EN ISO 29481-2:20XX

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;
```

## **В.3 Шаблони**

### **В.3.1 AppendixGroup**

**Атрибути:** *id*

**Елементи:** *state, dateLaMu, userLaMu*

**Посилання:** *group*

Таблиця зіставлення відповідних взаємовідношень n:m між додатками та групами.

#### *Приклад*

Приклад на рівні повідомлення:

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

Асоціативно пов'язана частина структури взаємодії:

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

### **В.3.2 AppendixTemplate**

**Атрибути:** *id*

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

**Посилання: *appendixGroup, message***

Реєстрування вкладених файлів.

**Приклад**

Приклад на рівні повідомлення:

```

<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>

```

**Асоціативно пов'язана частина структури взаємодії:**

```

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

```

### **V.3.3 ComplexElementTemplate**

**Атрибути:** id

### **V.3.4 GroupTemplate**

**Атрибути:** id

**Елементи:** *name, description, creationDate, startDate, endDate, state, dateLaMu, userLaMu, versionNo*

**Посилання:** *transaction*

Групування додатків повідомлення для відновлення асоціативно пов'язаних документів.

#### *Приклад*

Приклад на рівні повідомлення:

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

**Асоціативно пов'язана частина структури взаємодії:**

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

```

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

```

### **V.3.5 MessageInTransactionTemplate**

**Атрибути:** id

**Елементи:** *identification, dateSend, dateRead, state, dateLaMu, userLaMu, initiatorToExecutor*

Уміщує у повідомленні фактичний тип *MessageInTransactionType* для відновлення стану робочого циклу транзакції.

### **V.3.6 MessageTemplate**

**Атрибути:** id

**Елементи:** *identification, dateSend, dateRead, state, dateLaMu, userLaMu, initiatingTransactionMessageID, initiatorToExecutor*

**Посилання:** *transaction, messageInTransaction, template*

Екземпляр *MessageType*. Ця сутність здійснює фактичне обмінювання інформацією між екземплярами *OrganisationTemplate* (організаціями).

*Приклад*

**Приклад на рівні повідомлення:**

```
<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>
```

**Асоціативно пов'язана частина структури взаємодії:**

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

```

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

```

### **V.3.7 OrganisationTemplate**

**Атрибути:** id

**Елементи:** *name, abbreviation, state, dateLaMu, userLaMu*

**Посилання:** *contactPerson, template*

Зазначає організацію, яка бере участь у проекті, ініціюючи або виконуючи *TransactionTemplate* (транзакцію).

*Приклад*

Приклад на рівні повідомлення:

```

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

```

## прДСТУ EN ISO 29481-2:20XX

```
<userLaMu>bapa</userLaMu>
<contactPerson>
  <StandardPersonType id="..." >
    ...
  </StandardPersonType>
</contactPerson>
</StandardOrganisationType>
```

### Асоціативно пов'язана частина структури взаємодії:

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

### **B.3.8 PersonInRole**

**Атрибути:** *id*

**Елементи:** *state, dateLaMu, userLaMu*

**Посилання:** *organization, contactPerson, role, substituting, successor*

Зазначає людину, яка виконує певну роль у певній організації.

#### *Приклад*

Приклад на рівні повідомлення:

```
<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>

```

**Асоціативно пов'язана частина структури взаємодії:**

```

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

```

### **B.3.9 PersonTemplate**

**Атрибути:** id

**Елементи:** *userName, name, state, dateLaMu, userLaMu*

Зазначає особу, яка бере участь у проєкті як виконавець певної ролі або як контактна особа певної організації.

#### *Приклад*

Приклад на рівні повідомлення:

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

Асоціативно пов'язана частина структури взаємодії:

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

### **B.3.10 ProjectTemplate**

**Атрибути:** id

**Елементи:** *name, description, startDate, endDate, state, dateLaMu, userLaMu*

Зазначає структуру, призначену для відповідного уміщення в ній проєкту.

#### *Приклад*

Приклад на рівні повідомлення:

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

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

Associated part of the interaction

#### Асоціативно пов'язана частина структури взаємодії:

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

### B.3.11 RoleTemplate

#### Атрибути: id

#### Елементи: *name, description, state, dateLaMu, userLaMu, category*

Зазначає роль, яку від імені організації виконує *PersonTemplate* (особа).

#### Приклад

#### Приклад на рівні повідомлення:

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

#### Асоціативно пов'язана частина структури взаємодії:

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

```
<dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
<userLaMu>bapa</userLaMu>
</RoleType>
```

### **V.3.12 TransactionPhaseTemplate**

**Атрибути:** id

**Елементи:** *name, description, dateReached, state, dateLaMu, userLaMu*

**Посилання:** *transaction*

Зазначає поточну фазу транзакції.

*Приклад*

Приклад на рівні повідомлення:

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

**Асоціативно пов'язана частина структури взаємодії:**

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

```

<endDate>2008-12-31T00:00:00Z</endDate>
<state>active</state>
<dateLaMu>2008-01-23T00:00:00Z</dateLaMu>
<userLaMu>bapa</userLaMu>
</TransactionPhaseType>

```

### **V.3.13 TransactionTemplate**

**Атрибути:** *id*

**Елементи:** *number, name, description, startDate, endDate, state, dateLaMu, userLaMu, result*

**Посилання:** *initiator, executor*

Транзакція, яка уможлиблює обмінювання повідомленнями *MessageTemplates* для виконання певних завдань.

*Приклад*

Приклад на рівні повідомлення:

```

<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>

```

**Асоціативно пов'язана частина структури взаємодії:**

```

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

```

## прДСТУ EN ISO 29481-2:20XX

```
<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>
```

### **V.4 Атрибути**

#### **V.4.1 id**

Унікальний код для ідентифікації екземпляра об'єкта в повідомленні.

**Примітка.** У документі XML значення ідентифікатора має бути унікальним. Отже, звичайний серійний номер мав би задовольняти цій умові. Відтак, немає жодної переконливої причини обмежувати значення ідентифікаторів лише GUID (хоча у більшості реалізованих інтерактивних середовищ застосовують саме GUID). Варто зазначити, що у стандарті XML (<http://www.w3.org/XML/>) встановлено вимогу, щоб ідентифікатор починався з літери чи позначки ' \_ '.

### **V.5 Елементи**

#### **V.5.1 Abbreviation**

Скорочення назви організації. Цей елемент призначено для використання як префікс, долучений до номера транзакції, для ідентифікації цієї транзакції.

**Примітка.** Елемент *abbreviation*. Елемент *abbreviation* уміщує скорочену назву організації. Один із варіантів його застосування – позначати транзакції (у поєднанні з порядковим номером), розпочаті цією організацією, для полегшення неформального спілкування між людьми – учасниками комунікації.

## B.5.2 category

Категорія для класифікації зазначеного екземпляра об'єкта.

### Приклад

Приклад на рівні повідомлення:

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

## B.5.3 creationDate

Дата створення конкретного екземпляра об'єкта в зазначеному повідомленні.

### Приклад

Приклад на рівні повідомлення:

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

## B.5.4 dateLaMu

Дата останнього змінення.

### Приклад

Приклад на рівні повідомлення:

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

## B.5.5 dateReached

Дата «dateReached» – це атрибут дати/часу елемента *TransactionPhaseTemplate*. Він означає дату/час, коли було досягнуто певної фази транзакції.

*Приклад*

Приклад на рівні повідомлення:

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

**B.5.6 dateRead**

Дата прочитання зазначеного повідомлення.

*Приклад*

Приклад на рівні повідомлення:

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

**B.5.7 dateSend**

Дата надсилання зазначеного повідомлення.

*Приклад*

Приклад на рівні повідомлення:

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

**B.5.8 description**

Опис зазначеного екземпляра об'єкта.

*Приклад*

Приклад на рівні повідомлення:

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



### **V.5.9 documentIdentification**

Унікальний номер чи посилання на документ або файл для ідентифікації документа.

Примітка. Згідно з цим стандартом можливо установлювати різницю між версіями документів та версіями файлів без будь-яких обмежень (див. V.5.13).

### **V.5.10 documentReference**

Посилання для ідентифікації файлу або документа.

### **V.5.11 documentVersion**

Версія документа або файлу.

### **V.5.12 endDate**

Дата, коли завершено застосовність конкретного екземпляра об'єкта в цьому повідомленні.

#### *Приклад*

Приклад на рівні повідомлення:

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

### **V.5.13 fileLocation**

Розташування файлу, переважно, Інтернет-адреса або шлях доступу до спільно використовуваного сервера.

#### *Приклад*

Приклад на рівні повідомлення:

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

**Примітка.** Згідно з цим стандартом можливо розрізнити версії документів та версії файлів без будь-яких обмежень (див. V.5.9).

### **B.5.14 fileType**

Тип файлу, переважно тип MIME (*Multipurpose Internet Mail Extensions*), що передбачає багатоцільові розширення для Інтернет-пошти.

#### *Приклад*

Приклад на рівні повідомлення:

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

### **B.5.15 fileVersion**

Версія файлу, збільшуване ціле число або дата.

#### *Приклад*

Приклад на рівні повідомлення:

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

### **B.5.16 identification**

Ідентифікація зазначеного повідомлення, орієнтована на людину-користувача.

#### *Приклад*

Приклад на рівні повідомлення:

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

### **B.5.17 initiatingTransactionMessageID**

Посилання на повідомлення, що належить до підпорядкованої транзакції, в якій було ініційовано це повідомлення.

### B.5.18 initiatorToExecutor

Напрямок зв'язку, застосовний до передавання зазначеного конкретного повідомлення.

#### Приклад

Приклад на рівні повідомлення:

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

У цьому прикладі екземпляр повідомлення *ExampleMessage* передаватимуть від В (виконавця) до А (ініціатора).

### B.5.19 language

Мова вмісту додатка.

#### Приклад

Приклад на рівні повідомлення:

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

### **V.5.20 name**

Іменування (STRING)

### **V.5.21 number**

Номер транзакції.

**Примітка.** Елемент *number*. Порядковий номер транзакції (див. також попередній пункт).

### **V.5.22 objectCode**

Забезпечує можливість пов'язання із зовнішнім кодовим позначенням/ індексом. Наприклад, структура декомпозиції робіт, пакети робочих документів або специфікації будівель.

### **V.5.23 result**

Результат зазначеної транзакції.

### **V.5.24 startDate**

Початкова дата застосування конкретного екземпляра об'єкта в зазначеному повідомленні.

#### *Приклад*

Приклад на рівні повідомлення:

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

### **V.5.25 state**

Поточний стан зазначеного екземпляра об'єкта.

#### *Приклад*

Приклад на рівні повідомлення:

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

&lt;/...&gt;

**B.5.26 userLaMu**

Користувач, який виконав останню зміну (варто звернути увагу на те, що це не є посиланням на екземпляр *PersonTypeInstance*).

*Приклад*

Приклад на рівні повідомлення:

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

**B.5.27 userName**

Ініціали користувача, наприклад три грама (послідовність із трьох літер).

*Приклад*

Приклад на рівні повідомлення:

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

**B.5.28 versionNo**

Номер версії екземпляра об'єкта, пов'язаного із цим проектом; після змінення цього екземпляра об'єкта зазначене поле також слід оновити.

*Приклад*

Приклад на рівні повідомлення:

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

## **V.6 Посилання**

### **V.6.1 appendixGroup**

Підрозділ структури для визначення певного додатка.

#### *Приклад*

Приклад на рівні повідомлення:

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

Вважають, що в асоціативно пов'язаній частині структури визначено тип *AppendixType* зі значенням «Appendix».

### **V.6.2 contactPerson**

Особа, пов'язана з об'єктом *PersonInRole* або із певною організацією.

#### *Приклад*

Приклад на рівні повідомлення:

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

Вважають, що в асоціативно пов'язаній частині структури визначено тип *OrganisationType* зі значенням «Organisation» та тип *PersonType* зі значенням «Person».

### **V.6.3 executor**

Об'єкт *PersonInRole*, який буде діяти як виконавець цієї транзакції.

### **V.6.4 group**

Загальна група для складання набору додатків.

#### *Приклад*

Приклад на рівні повідомлення:

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

Вважають, що в асоціативно пов'язаній частині структури визначено тип *GroupType* зі значенням «Group».

### **V.6.5 initiator**

Об'єкт *PersonInRole*, який буде діяти як ініціатор цієї транзакції.

### **V.6.6 message**

Повідомлення, що містить конкретний додаток.

#### *Приклад*

Приклад на рівні повідомлення:

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

Вважають, що в асоціативно пов'язаній частині структури визначено тип *AppendixType* зі значенням «Appendix» та тип *MessageType* зі значенням «Message».

### **B.6.7 messageInTransaction**

Посилання на позицію зазначеного повідомлення в інформаційному потоці транзакції.

### **B.6.8 organization**

Організація, співвідносна з об'єктом *PersonInRole*.

#### *Приклад*

Приклад на рівні повідомлення:

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

Вважають, що в асоціативно пов'язаній частині структури визначено тип *OrganisationType* зі значенням «Organisation».

### **B.6.9 role**

Посилання на роль, яку від імені організації виконує *PersonTemplate* (особа).

### **B.6.10 substituting**

Об'єкт *PersonInRole*, авторизований, щоб надсилати повідомлення від імені зазначеної особи.

**Примітка.** Посилання *substituting*. Посилання на формальний об'єкт *PersonInRole* для цієї транзакції, що дозволяє іншій особі діяти за відсутності особи – формального виконавця ролі *PersonInRole*. Обидві особи мають бути пов'язані посиланням з одним і тим самим типом ролі.



### **B.6.11 successor**

Наступник іншої особи в певній ролі.

### **B.6.12 transaction**

Транзакція, в якій уміщено певну групу, повідомлення чи фазу транзакції.

#### *Приклад*

Приклад на рівні повідомлення:

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

Вважають, що в асоціативно пов'язаній частині структури визначено тип *MessageType* зі значенням «Message» та тип *TransactionType* зі значенням «Transaction».

## **ДОДАТОК С**

(довідковий)

### **ПРИКЛАД СПРОЩЕНОЇ КАРТИ ВЗАЄМОДІЇ ДЛЯ ПРОЕКТНОГО БЮРО**

#### **С.1 Загальні положення**

У наведеному прикладі створення структури взаємодії сферу взаємодії було обмежено представленням її спрощеного варіанту для проектного бюро (див. приклад у розділі 4). Процес взаємодії устанавлюють у структурі визначенням:

- ролей;
- транзакцій;
- повідомлень у транзакціях;
- послідовності повідомлень у транзакції;
- елементів даних у повідомленнях.

#### **С.2 Ролі та транзакції**

Щоб проілюструвати масштаби взаємодії у наведеному прикладі, дії з комунікації зображено на «карті взаємодії», де ролі взаємопов'язано між собою за допомогою транзакцій.

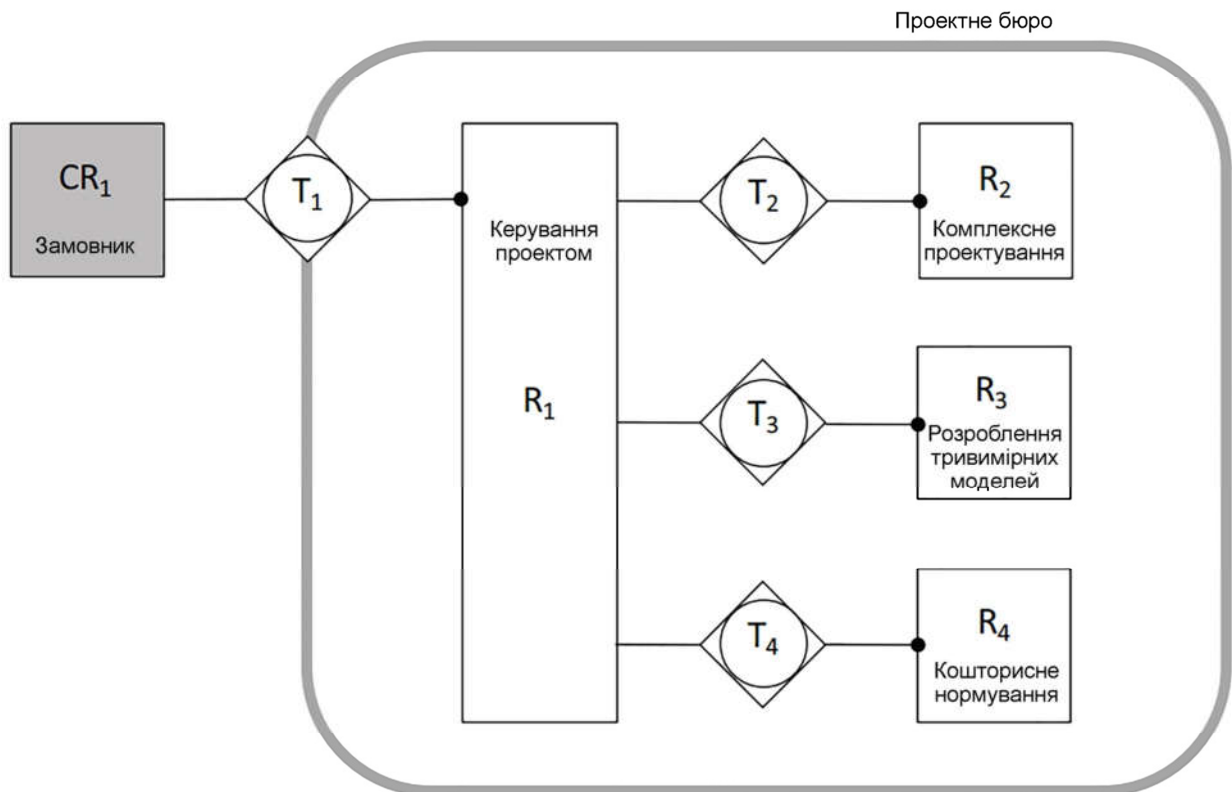
Ролі:

- CR<sub>1</sub>: Замовник;
- R<sub>1</sub>: Керування проектом;
- R<sub>2</sub>: Комплексне проектування;
- R<sub>3</sub>: Розроблення тривимірних моделей;
- R<sub>4</sub>: Кошторисне нормування.

Нижче викладено зведену таблицю ролей, пов'язаних із взаємодією у транзакції.

**Таблиця С.1** – Таблиця ролей у транзакціях для спрощеного проектного бюро

Тип транзакції	Ініціювальна роль	Виконавча роль
T <sub>1</sub> Доставляння поекту	CR <sub>1</sub> Замовник	R <sub>1</sub> Керування проектом
T <sub>2</sub> Доставляння специфікації комплексного проекту	R <sub>1</sub> Керування проектом	R <sub>2</sub> Комплексне проектування
T <sub>3</sub> Доставляння тривимірної моделі	R <sub>1</sub> Керування проектом	R <sub>3</sub> Розроблення тривимірних моделей
T <sub>4</sub> Доставляння кошторису	R <sub>1</sub> Керування проектом	R <sub>4</sub> Кошторисне нормування



**Рисунок С.1** – Карта взаємодії, в якій визначено відповідні типи ролей і типи транзакцій

## C.2.1 RoleType

Ролі визначають у структурі як типи *RoleTypes*. Кожен тип ролі може бути визначено за дотримання таких умов:

- зазначення ID (ідентифікація *roleType*);
- наведення дійсного значення ідентифікатора XML ID (кваліфіковане ім'я); щодо нього існують певні обмеження:
  - він має бути унікальним у цій моделі;
  - деякі символи не допустимі ('', '\$ ', '#', '@', ...);
  - його записують без пробілів,
  - його не можна починати з цифри;
- наведення опису (специфікація *roleType*),
- зазначення обов'язків (обсяг, завдання, забезпечення та зворотний зв'язок);
- наведення метаданих (остання зміна, довідкова інформація тощо).

У наведеному прикладі структури було визначено чотири ролі (див. приклад нижче, представлений у форматі XML):

```
<RoleType id="CR1_Client" >
  <description>CR1: Client</description>
  <startDate>2010-10-29T00:00:00.000+02:00</startDate>
  <endDate>2099-10-29T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-29T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <code></code>
  <responsibilityScope>Is responsible for the realisation of the project in
terms of time, quality and cost</responsibilityScope>
  <responsibilityTask></responsibilityTask>
```

```

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

```

## прДСТУ EN ISO 29481-2:20XX

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

### С.3 Повідомлення у транзакції

Транзакція містить набір повідомлень, обмінювання якими відбувається для досягнення певної цілі. Транзакцією обумовлено також ролі учасників, момент у життєвому циклі та послідовність доставлення повідомлень (за потреби).

#### С.3.1 TransactionType

Транзакції визначають у структурі як типи *TransactionTypes*. Кожен тип транзакції може бути визначено за дотримання таких умов:

- зазначення ID (ідентифікація *TransactionType*);
- наведення опису (специфікація *TransactionType*),
- зазначення метаданих (результат *TransactionType*, *helpinfo* тощо),
- типи *RoleTypes*, які залучено до *TransactionType*.

У наведеному прикладі структури було визначено чотири типи *TransactionTypes* (див. приклад нижче, представлений у форматі XML):

- T<sub>1</sub>: Доставка проекту
- T<sub>2</sub>: Доставка специфікації комплексного проекту
- T<sub>3</sub>: Доставка тривимірної моделі
- T<sub>4</sub>: Доставка кошторису

```

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

```

## прДСТУ EN ISO 29481-2:20XX

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



```

<helpInfo></helpInfo>
<code></code>
<result>Cost calculation is delivered</result>
<initiator>
    <RoleTypeRef idref="R1_Project_Leading"/ >
</initiator>
<executor>
    <RoleTypeRef idref="R4_Cost_Engineering"/ >
</executor>
</TransactionType>

```

В IDM установлюють відмінність між роллю, яка полягає у поданні запиту (ініціатор), та роллю, яка реалізує цей запит (виконавець). Потрібно, щоб у транзакції була лише одна ініціувальна роль і лише одна виконавча роль. Наприклад, тип *TransactionType* зі значенням «T3 Deliver 3D model» може бути ініційовано лише роллю R1 «Керування проектом» та виконано роллю R3 «Розроблення тривимірних моделей».

За допомогою транзакцій передану інформацію долучають до контексту процесу.

### C.3.2 TransactionPhaseType

Щоб зазначити стан взаємодії в межах транзакції, фази можуть бути пов'язані з типами *TransactionType*.

Із типами *TransactionType* може бути пов'язано фази, що уможливають визначення стану взаємодії в межах транзакції. Насправді, типи *TransactionPhaseType* у межах *TransactionType* являють собою шаблон транзакції. Визначення *TransactionPhaseType* передбачають ідентифікацію типу *TransactionPhaseType* та деяких метаданих. Кількість фаз та їх описи не обмежено.

У наведеному прикладі було визначено типи фаз *TransactionPhaseType*, що пов'язані з зазначеними на рисунку 2 у розділі 3, зокрема:

- очікування результату (*desired result*);
- надання запиту щодо результату (*result requested*);
- прийняття зобов'язання щодо результату (*result promised*);
- досягнення результату (*result produced*);
- доставляння результату (*result stated*);
- приймання результату (*result accepted*).

У наведеному нижче прикладі у форматі XML представлено визначення TransactionPhaseType:

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

```

<state>active</state>
<dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
<userLaMu>Ronald</userLaMu>
<language></language>
<category></category>
<helpInfo></helpInfo>
<code>-</code>
</TransactionPhaseType>
<TransactionPhaseType id="Result_Promised" >
  <description>Result promised</description>
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <code>-</code>
</TransactionPhaseType>
<TransactionPhaseType id="Result_Requested" >
  <description>Result requested</description>
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <code>-</code>
</TransactionPhaseType>
<TransactionPhaseType id="Result_Stated" >
  <description>Result stated</description>
  <startDate>2010-11-04T00:00:00.000+01:00</startDate>
  <endDate>2099-11-04T00:00:00.000+01:00</endDate>
  <state>active</state>
  <dateLaMu>2010-11-04T00:00:00.000+01:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <language></language>
  <category></category>
  <helpInfo></helpInfo>
  <code>-</code>

```

### C.3.3 MessageType

Повідомлення являє собою заповнену інформаційну модель з вмістом даних. Тип *MessageType* визначає структуру та забезпечує елементи вмісту повідомлення.

У кожному повідомленні є групи сегментів (складені елементи). Ці складені елементи визначають за типами *ComplexElementTypes*. У межах кожного сегмента може бути розміщено сукупність складених елементів та/або одиночних елементів. Одиночні елементи визначають за типами *SimpleElementTypes*.

У наведеному нижче прикладі у форматі XML представлено визначення типів *MessageTypes*, графічне зображення яких наведено на рисунку 5 у розділі 3:

- запит щодо розробки тривимірної моделі (*request for 3D model*);
- розробку виконано та направлено запит щодо схвалення (*work done and request approval*);
- запит щодо коригування (*request adjustments*);
- розробку схвалено (*work approved*);
- розробку не схвалено (*work not approved*).

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

```

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

```

## прДСТУ EN ISO 29481-2:20XX

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

### C.3.4 MessageInTransactionType

Транзакція містить набір повідомлень, обмін якими здійснюють для досягнення певної цілі. Визначення *MessageInTransactionType* передбачає пов'язання типів *MessageTypes* із типами *TransactionTypes*. Один і той самий *MessageType* може бути пов'язано з усіма визначеними *TransactionTypes*. Так само можна пов'язати один і той самий *MessageType* декілька разів (навіть необмежену кількість разів) з одним і тим самим *TransactionType*.

Кожен тип *MessageInTransactionType* може бути визначено за дотримання таких умов:

- зазначення ID (ідентифікатор *MessageInTransactionType*);
- установлення напрямку, у якому має бути надіслано повідомлення: від ініціатора до виконавця – як дійсного (= *true*); від виконавця до ініціатора – як помилкового (= *false*);
- наведення метаданих *MessageInTransactionType*,
- установлення посилання на *MessageType* (позицію якого визначено в *MessageInTransactionType*);
- установлення посилання на тип (-и) *TransactionType* для пов'язання із зазначеними типами *MessageType*;
- установлення посилання на фазу *TransactionPhase*, до якої переходить процес (після надсилання *MessageType* у *TransactionType*, згідно з визначеним типом *MessageInTransactionType*),
- установлення посилання на тип *GroupType*.

У наведеному нижче прикладі у форматі XML представлено визначення двох типів *MessageInTransactionTypes* із зазначеного прикладу з доставляння розробки тривимірної моделі. Для кращого розуміння цей приклад варто розглядати, ознайомившись із

рисунком 5 (у попередній версії частини 2 цього стандарту ISO/WD 29481-2, щодо IDM, див. Карту взаємодії у розділі 3).

У наведеному нижче прикладі в XML показано, що тип *TransactionType* T3 запускають за допомогою *MessageType* «Запит щодо розробки тривимірної моделі» (*Request for 3D model*). Це повідомлення є першим у транзакції, оскільки перед ним не зазначено жодного іншого *MessageInTransactionType*, що визначено в межах *MessageInTransaType* зі значенням «*MessageInTransaction1*». Напрямок *MessageInTransaction1* зазначено від ініціатора до виконавця (значення «= *true*»).

Другий тип *MessageInTransactionType*, зі значенням «*MessageInTransaction2*», пов'язаний із типом *MessageType* «Розробку виконано та направлено запит щодо схвалення» (*Work done and request approval*). Згідно з рисунком 5 показано, що це повідомлення у послідовності має бути наступним після двох попередніх повідомлень:

- Запит щодо розробки тривимірної моделі (*Request for 3D model*);
- Запит щодо коригування (*Request adjustments*).

Ці два попередні повідомлення було визначено як типи *MessageInTransactionTypes* (номери 1 та 3) у межах *MessageInTransactionType2*. Напрямок *MessageInTransaction1* зазначено від виконавця до ініціатора (значення «= *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

До повідомлень може бути долучено пов'язані посиланнями вкладені файли. У вигляді вкладення вимогу щодо обміну може бути

прДСТУ EN ISO 29481-2:20XX

надіслано виконавчій ролі, а результат роботи (для долучення до BIM) – ініціювальній ролі.

Визначення типу *AppendixType* забезпечує установлення обмежень, застосовних до елементних інформаційних об'єктів, пов'язаних із документами, додаванням метаданих до вкладених у повідомлення документів.

У наведеному нижче прикладі у форматі XML показано визначення типу *AppendixType*. Метадані визначено за допомогою типів *SimpleElementTypes* у межах *ComplexElementType* зі значенням «*AppendixComplexElement*».

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

### **C.3.6 Обмеження, застосовні до вмісту повідомлень**

Третій і завершальний етап визначення структури взаємодії полягає у визначенні компонентів у межах типів *MessageTypes*, щоб:

- структурувати типи *MessageTypes*;
- установити обмеження щодо введення вмісту.

Визначення *MessageType* вже містить складові елементи повідомлення. Ці складові елементи зазначають як типи *ComplexElementTypes*.

*ComplexElementTypes* містять у своєму складі типи *SimpleElementTypes* (або, можливо, також інші *ComplexElementTypes*). Типи *SimpleElementTypes* пов'язують із типами *UserDefinedTypes* для визначення обмежень щодо введення вмісту. Графічне зображення цієї структури наведено на рисунку нижче.

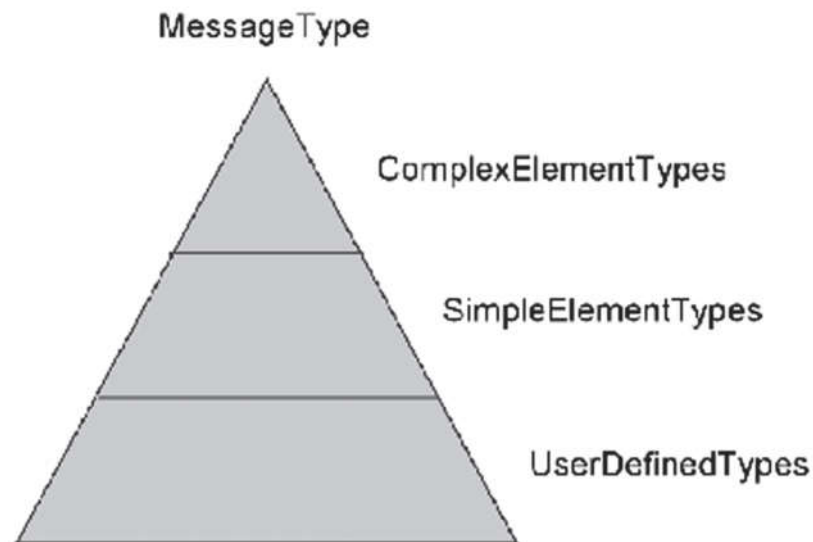


Рисунок С.2 – Компоненти моделі повідомлення

### С.3.7 Типи *ComplexElementType*

За допомогою визначення типів *ComplexElementType* забезпечують:

- обмеження елементних інформаційних об'єктів,
- використання механізмів ієрархії визначення типів (*Type Definition Hierarchy*) у XML для отримання похідного складеного типу з іншого простого чи складеного типу;
- визначення функціональної придатності елементів до перевіряння на відповідність схемі даних (*post-schema-validation info set*);
- обмеження можливості отримання додаткових типів, похідних від заданого складеного типу;

– управління дозволами на заміщення, наприклад, елементів похідного типу елементами, зазначеними в моделі вмісту даних як складений тип.

З огляду на наведений вище приклад типу *MessageType* «Запит щодо розробки тривимірної моделі» (*Request for 3D Model*), пов'язаного з транзакцією ТЗ «Доставка розробки тривимірної моделі» (*Deliver 3D model*), цей тип *MessageType* містить такі складені типи *ComplexElementTypes*:

- *resultRequestedComplexElement*;
- *workIdentificationComplexElement*;
- *workSpecificationComplexElement*;
- *remarksComplexElement*;
- *deadlineComplexElement*.

Нижче у форматі XML представлено тип *ComplexElementType* зі значенням «*WorkSpecification*». Типом *ComplexElementType* обумовлено його ідентифікацію, посилання на прості елементи, за допомогою яких передають уміст до *complexElementType*, а також його метадані.

У прикладі представлення в XML типу *ComplexElementType* зі значенням «*WorkSpecificationComplexElement*» показано, що *ComplexElementType* містить два типи *SimpleElementType*:

- *ScopeOfWork*;
- *CostEstimation*.

```
<ComplexElementType id="WorkSpecificationComplexElement" >  
  <description>Work specification</description>  
  <startDate>2010-10-01T00:00:00.000+02:00</startDate>  
  <endDate>2099-10-01T00:00:00.000+02:00</endDate>  
  <state>active</state>  
  <dateLaMu>2010-10-01T00:00:00.000+02:00</dateLaMu>
```

```

<userLaMu>Ronald</userLaMu>
<language></language>
<category></category>
<helpInfo></helpInfo>
<simpleElements>
    <SimpleElementTypeRef idref="ScopeOfWork" / >
    <SimpleElementTypeRef idref="CostEstimation" / >
</simpleElements>
</ComplexElementType>

```

### C.3.8 Типи SimpleElementType

За допомогою визначення типів *SimpleElementType* забезпечують:

- установлення їх ідентифікації;
- установлення «простору значень» та обмежень щодо вхідних даних.

У наведеному нижче прикладі в форматі XML показано визначення одного з типів *SimpleElementType*, зокрема, «CostEstimation». Цей тип *SimpleElementType* є частиною типу *ComplexElementType* «WorkSpecificationComplexElementType» (див. попередній приклад).

```

<SimpleElementType id="CostEstimation" >
    <description>Cost estimation</description>
    <interfaceType/>
    <state>active</state>
    <dateLaMu>2010-12-10T12:36:02.527+01:00</dateLaMu>
    <userLaMu>Ronald</userLaMu>
    <language/>
    <category/>
    <helpInfo>Indication of costs </helpInfo>
    <valueList/>
    <userDefinedType>
        <UserDefinedTypeRef idref="EURO" / >
    </userDefinedType>
</SimpleElementType>

```

### C.3.9 Типи *UserDefinedTypes*

Типи *UserDefinedTypes* використовують для визначення обмежень щодо введення вмісту (визначенням типів даних *dataType*). У представленні XML дозволено усі можливі обмеження для визначення типів *UserDefinedTypes* у структурі взаємодії. У наведеному вище прикладі було використано лише типи даних «STRING» та «EURO».

У наведеному нижче прикладі в XML показано визначення типу *UserDefinedType*, за допомогою якого забезпечують:

- ідентифікацію;
- метадані;
- тип *baseType*;
- обмеження XSD (для визначення більш конкретного обмеження

або для визначення типів переліків).

```
- <UserDefinedType id="EURO" >
  <description>Euro</description>
  <state>active</state>
  <dateLaMu>2010-12-10T12:35:19.485+01:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <baseType>DECIMAL</baseType>
  <xsdRestriction><xs:fractionDigits value="2"/ > </xsdRestriction >
  <helpInfo>amounts in Euro, specified in two decimals</helpInfo>
</UserDefinedType>
- <UserDefinedType id="STRING" >
  <description>Character strings</description>
  <state>active</state>
  <dateLaMu>2010-10-01T00:00:00</dateLaMu>
  <userLaMu>Ronald</userLaMu>
  <baseType>STRING</baseType>
  <xsdRestriction />
  <language />
  <helpInfo>The string datatype represents character strings in XML. The
  'value space' of string is the set of finite-length sequences of
  characters.</helpInfo>
```

</UserDefinedType>

### **С.3.10 Прикінцеве положення**

Для створення структури взаємодії було визначено всі типи елементів. Щоб перевірити правильність структури, потрібно визначити її відповідність до схеми XSD. Наступним етапом є забезпечення підтримки програмних рішень структури взаємодії для отримання схеми інтерактивного середовища. Потрібні для цього заходи описано в 4.8.

## ДОДАТОК D

(довідковий)

### ПРИНЦИП ЗАСТОСУВАННЯ АЛГОРИТМУ ПРОМОТОРА

#### D.1 Принципи алгоритму промотора

Промотор є алгоритмом, який перетворює структуру взаємодії на схему взаємодії. Оскільки структуру взаємодії записують у форматі XML, а схему взаємодії – у XSD, можна зробити висновок, що промотор утворює абстрактні елементи даних структури взаємодії у межах структури взаємозв'язків та сутностей схеми взаємодії.

Базовий алгоритм промотора перетворює у структурі взаємодії всі елементи даних «\*Type» (*RoleType*, *TransactionType*, *MessageType*, *ComplexElementType*, *SimpleElementType* тощо) у елементи складеного типу XSD. Звичайно, потрібно дотримуватися певних правил, щоб гарантувати умови, за яких отриманий XSD буде коректною схемою XML. Наприклад, значення атрибута ID елемента даних «\*Type» інтерпретують як ім'я аналогічного йому складеного елемента XSD. Внаслідок цього у структурі взаємодії атрибут ID не виглядає, як ID = «Role003», а здебільшого набуває вигляду, подібного до ID = «Project\_Manager».

Схему XSD організовано так, що типи *MessageType* є основними складеними елементами для структуризації вмісту повідомлення на рівні фактичного обміну в транзакції. Таке повідомлення має містити достатньо інформації, щоб простежити його фактичну позицію в загальному потоці повідомлень та визначити наступні повідомлення, які може бути відправлено з цієї позиції.



Щоб створити правильні типи атрибутів та взаємозв'язків для отриманої схеми XSD, промотор звертається до файлу, в якому описано, який шаблон і для якого типу складених елементів треба використовувати (файл шаблонів).

## **БІБЛІОГРАФІЯ**

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

### **НАЦІОНАЛЬНЕ ПОЯСНЕННЯ**

1 Ян Л. Г. Дітц (2006). Онтологія підприємства: теорія і методологія, Springer; 1 видання, 240 сторінок, ISBN: 3540291695

**Код згідно з НК 004 91.010.01**

**Ключові слова:** будівельне інформаційне моделювання, будівельний об'єкт, доставляння інформації, інтерактивне середовище, настанова, обмін інформацією, повідомлення, роль, транзакція, структура взаємодії, схема взаємодії, формат

Генеральний директор  
ТОВ «Укрінсталькон  
ім. В.М. Шимановського»,  
заслужений діяч науки і техніки України,  
член-кореспондент НАНУ, д.т.н., проф.

О. В. Шимановський

Заступник генерального директора з  
наукової роботи, д.т.н., проф.

В. М. Гордєєв

Заступник генерального директора з  
науково-технічної політики,  
заступник голови ТК 301  
(науковий керівник розробки)

В. П. Адріанов

Завідувач відділу

О. І. Кордун