ФЕДЕРАЛЬНОЕ АГЕНТСТВО
ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО/ТС 10303-25— 2012

Системы автоматизации производства и их интеграция

ПРЕДСТАВЛЕНИЕ ДАННЫХ ОБ ИЗДЕЛИИ И ОБМЕН ЭТИМИ ДАННЫМИ

Часть 25

Методы реализации. Связь EXPRESS с XMI

ISO/TS 10303-25:2005
Industrial automation systems and integration — Product data representation and exchange — Part 25: Implementation methods. EXPRESS to XMI binding (IDT)

Издание официальное

Москва
Стандартинформ
2013

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения»

Сведения о стандарте

1 ПОДГОТОВЛЕН Федеральным государственным автономным научным учреждением «Центральный научно-исследовательский и опытно-конструкторский институт робототехники и технической кибернетики» на основе собственного аутентичного перевода на русский язык международного документа, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 459 «Информационная поддержка жизненного цикла изделий»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 03 сентября 2012 г. № 267-ст

4 Настоящий стандарт идентичен международному документу ИСО/ТС 10303-25:2005 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 25. Методы реализации. Связь EXPRESS с XMI» (ISO/TS 10303-25:2005 «Industrial automation systems and integration — Product data representation and exchange — Part 25: Implementation methods. EXPRESS to XMI binding»).

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

5 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемом информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет

© Стандартинформ, 2013

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения и сокращения . . . 2
4 Соответствие . . . 3
4.1 Соответствие препроцессора . . . 3
4.2 Соответствие обменного документа . . . 4
5 Отображение конструкций языка EXPRESS на метамодель обмена UML . . . 4
5.1 Неотображаемые понятия языка EXPRESS . . . 4
5.2 Общие требования к отображению . . . 5
5.3 Отображения для простых типов данных языка EXPRESS . . . 6
5.4 Отображения для объявлений EXPRESS-схем и спецификаций интерфейсов . . . 6
5.5 Отображения для объектных типов данных языка EXPRESS . . . 7
5.6 Отображения для объявлений подтипов данных языка EXPRESS . . . 7
5.7 Отображения для явных и инверсных атрибутов языка EXPRESS . . . 8
5.8 Отображения для вычисляемых атрибутов языка EXPRESS . . . 10
5.9 Отображения для определенных типов данных языка EXPRESS . . . 10
5.10 Отображения для агрегированных типов данных языка EXPRESS . . . 11
Приложение А (обязательное) Регистрация информационного объекта . . . 14
Приложение В (справочное) Примеры схем, DTD, UML-диаграмм и XMI-файлов . . . 14
Приложение С (справочное) Представление на языке EXPRESS частей метамодели обмена UML . . . 25
Приложение D (справочное) Выдержки из Спецификации UML 1.4 . . . 29
Приложение Е (справочное) Краткое изложение отображения . . . 34
Приложение F (справочное) Техническое обсуждение . . . 37
Приложение G (справочное) Сценарии применения . . . 39
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов и документов ссылочным национальным стандартам Российской Федерации . . . 41
Библиография . . . 42

Введение

Стандарты комплекса ИСО 10303 распространяются на компьютерное представление информации об изделиях и обмен данными об изделиях. Их целью является обеспечение нейтрального механизма, способного описывать изделия на всем протяжении их жизненного цикла. Этот механизм применим не только для обмена файлами в нейтральном формате, но является также основой для реализации и совместного доступа к базам данных об изделиях и организации архивирования.

Стандарты комплекса ИСО 10303 представляют собой набор отдельно издаваемых стандартов (частей). Стандарты данного комплекса относятся к одной из следующих тематических групп: «Методы описания», «Методы реализации», «Методология и основы аттестационного тестирования», «Интегрированные обобщенные ресурсы», «Интегрированные прикладные ресурсы», «Прикладные протоколы», «Комплекты абстрактных тестов», «Прикладные интерпретированные конструкции» и «Прикладные модули». Полный перечень стандартов комплекса ИСО 10303 можно получить на сайте http://www.tc184-sc4.org/titles/STEP_Titles.htm. Настоящий стандарт входит в тематическую группу «Методы реализации». Он подготовлен подкомитетом SC4 «Производственные данные» Технического комитета 184 ИСО «Системы автоматизации производства и их интеграция».

Настоящий стандарт устанавливает связь языка EXPRESS с XMI.

Рабочая группа по развитию стандартов объектного программирования [Object Management Group (OMG)] стандартизировала спецификацию обмена метаданными XML [XML Metadata Interchange specification (XMI)], объединяющую стандарты OMG, в которых определены унифицированный язык моделирования [Unified Modeling Language (UML)] и метаобъектные средства [Meta-Object Facility (MOF)], со стандартом Консорциума по разработке стандартов и протоколов Интернета [World Wide Web Consortium (W3C)], определяющим расширяемый язык разметки [Extensible Markup Language (XML)] [2]. XMI предоставляет механизм для обмена метаданными между средствами моделирования на базе UML и репозиториями метаданных на базе MOF. Кроме того, OMG стандартизировала совместимый с XML формат обмена для UML, установив тем самым лексическое представление UML-моделей, основанных на унифицированной метамодели UML. Данное лексическое представление обладает, помимо всего прочего, способностью обмениваться информацией о типах данных, информацией о классах (или объектах), группировками классов, обеспечивающих пространства имен для классов (или схем), связями и наследованиями между классами (или подтипами).

Настоящий стандарт определяет отображение конструкций, представленных на языке EXPRESS, на метамодель UML. Поскольку стандарт XMI определяет XML-представление конструкций метамодели UML, унифицированное отображение EXPRESS-конструкций на UML-конструкции поддерживает XMI-представление EXPRESS-схем. EXPRESS-схемы, являющиеся спецификациями данных, отображаются на понятия метамодели UML, присутствующие в статических структурных диаграммах UML. Настоящий стандарт не определяет отображение всех EXPRESS-конструкций на метамодель UML, так как данная метамодель не поддерживает все соответствующие EXPRESS-понятия. Установленное в настоящем стандарте отображение является односторонним отображением из языка EXPRESS на метамодель обмена UML. Данные ограничения делают отображение непригодным для общей взаимозаменяемости EXPRESS-схем и UML-моделей для целей информационного моделирования.

Пользователям настоящего стандарта необходимо детальное знание языка EXPRESS в редакции 1 и языка OMG UML 1.5.

На рисунке 1 показан пример использования настоящего стандарта, относящийся к реализации прикладного протокола ИСО 10303. В данном примере участвуют:

- специалист по информационному моделированию в формате STEP, разрабатывающий прикладной протокол ИСО 10303;
- реализация настоящего стандарта, представляющая собой программное приложение, реализующее настоящий стандарт;
- специалист по UML-моделированию, использующий инструментарий UML-моделирования для формирования спецификации системы программного обеспечения, исходя из UML-модели, сгенерированной реализацией настоящего стандарта;
- программа UML-моделирования, генерирующая базовый Java-код из UML-модели;
- разработчик программы на языке Java, выполняющий разработку системы программного обеспечения на основании Java-кода, автоматически сгенерированного программой UML-моделирования.

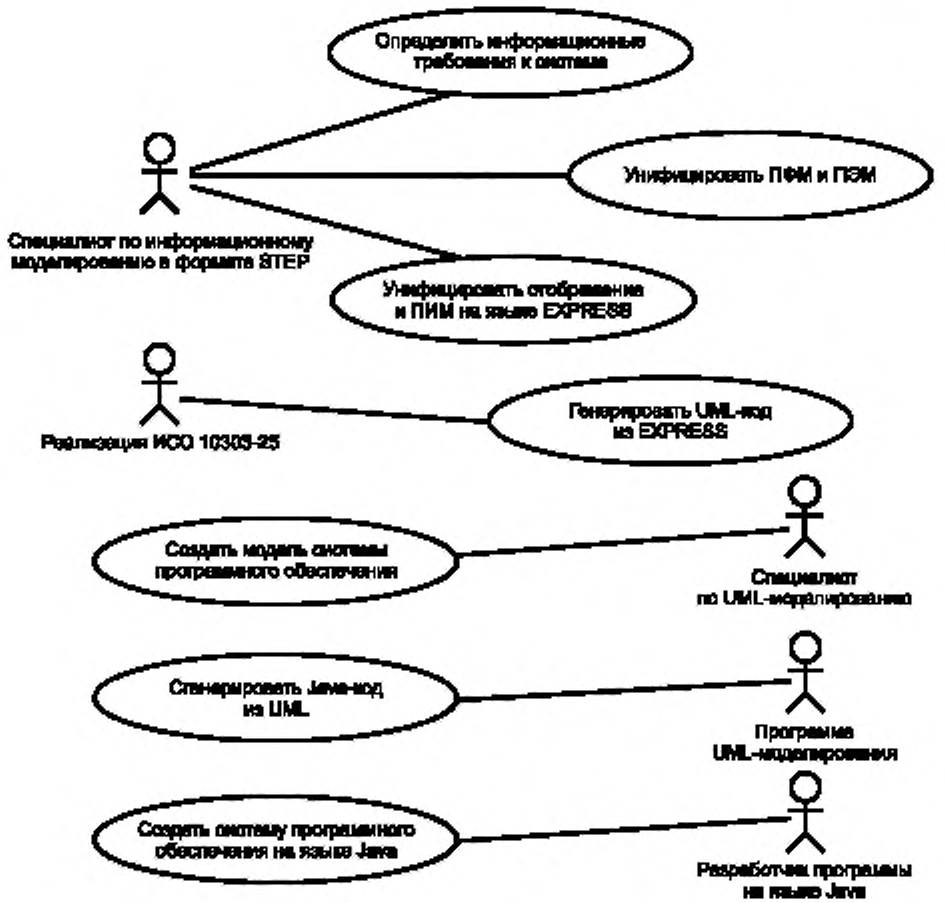

Рисунок 1 — Реализация прикладного протокола

На рисунке 2 показана концепция, положенная в основу настоящего стандарта, целью которой является обеспечение возможности использования средств разработки программного обеспечения на базе UML, если исходная модель данных написана на языке EXPRESS.

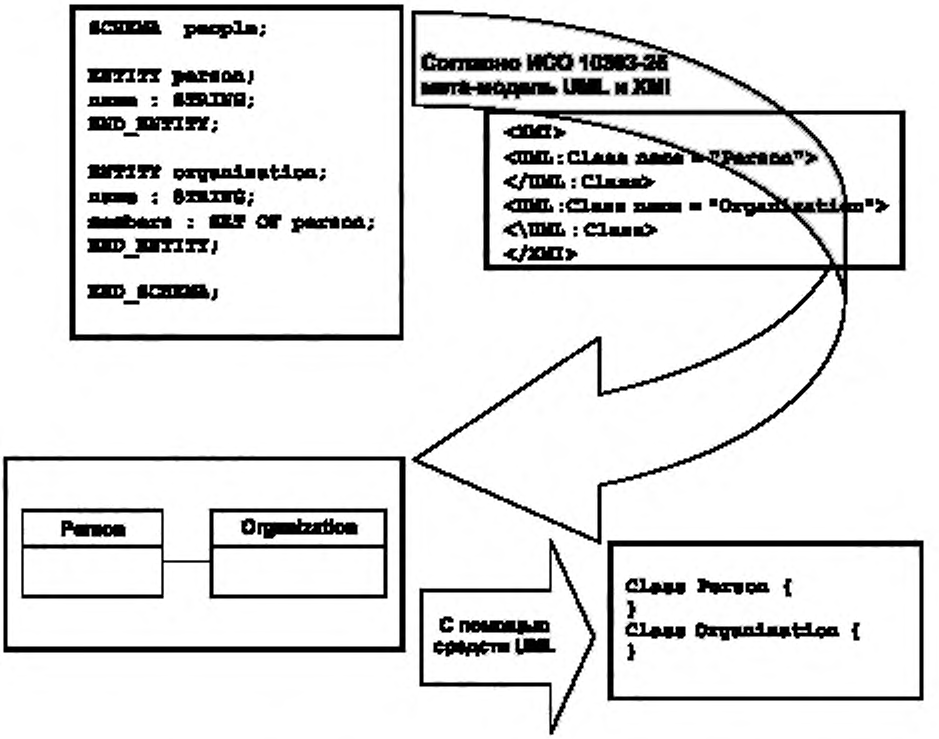

Рисунок 2 — Понятия, лежащие в основе связи EXPRESS с XMI

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы автоматизации производства и их интеграция

ПРЕДСТАВЛЕНИЕ ДАННЫХ ОБ ИЗДЕЛИИ И ОБМЕН ЭТИМИ ДАННЫМИ

Часть 25

Методы реализации. Связь EXPRESS с XMI

Industrial automation systems and integration. Product data representation and exchange. Part 25. Implementation methods. EXPRESS to XMI binding

Дата введения — 2013 — 05— 01

1 Область применения

Настоящий стандарт определяет отображение конструкций языка EXPRESS на метамодель обмена UML для обеспечения обмена данными, соответствующего стандарту XMI. Данное отображение является односторонним отображением из языка EXPRESS на метамодель обмена UML. Данные ограничения делают такое отображение непригодным для полного отображения EXPRESS-схем на UML-модели, необходимого для информационного моделирования.

Примечание — В общем случае обмен в соответствии со стандартом XMI является обменом информацией, представляющей UML-модель (т. е. схему), между инструментальными средствами UML-моделирования программных систем, репозиториями метаданных UML или другими подобными приложениями.

Требования настоящего стандарта распространяются на отображение конструкций языка EXPRESS на метамодель обмена UML для обеспечения обмена данными, соответствующего стандарту XMI.

Требования настоящего стандарта распространяются на:

- отображение конструкций языка EXPRESS на метамодель обмена UML для иных целей, кроме обмена с использованием XMI;
- отображение конструкций языка EXPRESS, которые не соответствуют метамодели обмена UML, включая:
  - глобальные и локальные правила;
  - отображение большинства ограничений на супертипы языка EXPRESS;
  - отображение выражений, функций, процедур и констант языка EXPRESS;
  - отображение явных атрибутов на языке EXPRESS, переобъявленных как производные атрибуты;
  - отображение комментариев на языке EXPRESS;
  - отображение понятий UML на язык EXPRESS.

Примечание — Большинство неотображаемых конструкций являются «правилами» (локальными правилами, глобальными правилами и некоторыми ограничениями на подтипы) и конструкциями, которые могут использоваться только в правилах. Хотя язык описания объектных ограничений (OCL), разработанный в OMG, предназначен для аналогичной цели в UML-моделях, лишь немногие понятия языка EXPRESS, используемые в правилах, имеют точные эквиваленты в OCL.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие международные стандарты и документы (для датированных ссылок следует использовать только указанное издание, для недатированных ссылок следует использовать последнее издание указанного документа, включая все поправки):

Издание официальное

ИСО/МЭК 8824-1:2002* Информационная технология. Абстрактная синтаксическая нотация версии 1 (АСН.1). Часть 1. Спецификация основной нотации (ISO/IEC 8824-1:2002, Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation)

ИСО 10303-1:1994 Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 1. Общие представления и основополагающие принципы (ISO 10303-1:1994, Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles)

ИСО 10303-11:1994** Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS (ISO 10303-11:1994, Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual)

ИСО 10303-28:2003*** Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 28. Методы реализации. Представления схем и данных EXPRESS на языке XML (ISO 10303-28:2003, Industrial automation systems and integration — Product data representation and exchange — Part 28: Implementation methods: XML representations of EXPRESS schemas and data)

Спецификация унифицированного языка моделирования OMG, версия 1.5, официальный документ Рабочей группы по развитию стандартов объектного программирования, март 2003 г. (цитирован 11.02.2004 г.). Доступен в Интернете по адресу: <gost.e-ecolog.ru> (OMG Unified Modeling Language Specification Version 1.5, Object Management Group Document formal/2003-03-01, March 2003 [cited 2004-02-11]. Available from World Wide Web: <gost.e-ecolog.ru>)

Спецификация обмена метаданными XML (XMI) OMG, версия 1.2, официальный документ Рабочей группы по развитию стандартов объектного программирования, январь 2002 г. (цитирован 11.02.2004 г.). Доступен в Интернете по адресу: <gost.e-ecolog.ru> (OMG XML Metadata Interchange (XMI) Specification Version 1.2, Object Management Group Document formal/2002-01-01, January 2002 [cited 2004-02-11]. Available from World Wide Web: <gost.e-ecolog.ru>)

Спецификация обмена метаданными XML (XMI) OMG, версия 2.0, официальный документ Рабочей группы по развитию стандартов объектного программирования, май 2003 г. (цитирован 11.02.2004 г.). Доступен в Интернете по адресу: <gost.e-ecolog.ru> (OMG XML Metadata Interchange (XMI) Specification Version 2.0, Object Management Group Document formal/2002-01-01, May 2003 [cited 2004-02-11]. Available from World Wide Web: <gost.e-ecolog.ru>)

Спецификация метаобъектных средств (MOF), версия 1.4, официальный документ Рабочей группы по развитию стандартов объектного программирования, апрель 2002 г. (цитирован 11.02.2004 г.). Доступен в Интернете по адресу: <gost.e-ecolog.ru> (Meta Object Facility (MOF) Specification Version 1.4, Object Management Group formal/2002-04-03, April 2002 [cited 2004-02-11]. Available from World Wide Web: <http://www.omg.org/cgi-bin/doc?formal/2003-05-02>)

3 Термины, определения и сокращения

3.1 Термины, определенные в ИСО 10303-1

В настоящем стандарте применены следующие термины:

- **прикладной протокол; ПП** (application protocol; AP);
- **данные** (data);
- **язык определения данных** (data specification language).

---

* Отменен. Действует ИСО/МЭК 8824-1:2008. Для однозначного соблюдения требований настоящего стандарта, выраженных в датированных ссылках, рекомендуется использовать только данный ссылочный стандарт.

** Отменен. Действует ИСО 10303-11:2004. Для однозначного соблюдения требований настоящего стандарта, выраженных в датированных ссылках, рекомендуется использовать только данный ссылочный стандарт.

*** Отменен. Действует ИСО/МЭК 10303-28:2007. Для однозначного соблюдения требований настоящего стандарта, выраженных в датированных ссылках, рекомендуется использовать только данный ссылочный стандарт.

3.2 Новый термин

В настоящем стандарте применен следующий термин с соответствующим определением:

3.2.1 **контекстная схема** (context schema): EXPRESS-схема, возможно включающая спецификации интерфейсов языка EXPRESS, которая определяет область применения отображения на язык UML.

3.3 Сокращения

В настоящем стандарте применены следующие сокращения:

AP — прикладной протокол (application protocol);
DTD — определение типа документа (Document Type Definition);
IDL — язык определения интерфейсов (Interface Definition Language);
MOF — метаобъектные средства (Meta Object Facility);
OCL — язык описания объектных ограничений (Object Constraint Language);
OMG — Рабочая группа по развитию стандартов объектного программирования (Object Management Group);
UDS — набор объявлений UML (UML declaration set);
XMI — обмен метаданными XML (XML Meta-data Interchange);
XML — расширяемый язык разметки (Extensible Markup Language).

4 Соответствие

Отображения из EXPRESS на метамодель UML являются отображениями на понятия статических структурных диаграмм (т. е. классов) UML. Любая ссылка на понятие UML является ссылкой на определение данного понятия, как это установлено в спецификации унифицированного языка моделирования OMG, версия 1.5, март 2003 г. Спецификация UML 1.5 включает метамодель обмена UML, на которой основаны отображения, определенные в настоящем стандарте.

Примечание — Отображения, определенные в настоящем стандарте, могут применяться во всех случаях, когда понятия, определенные в предыдущих версиях UML (версии 1.3 [3] и 1.4 [4]) эквивалентны понятиям, определенным в версии 1.5.

Для того чтобы выполнить представление, соответствующее XMI, используют метамодель обмена UML 1.5 совместно с правилами, определенными в спецификации XMI, для генерации стандартного формата, используя который, можно обмениваться UML-моделями. В стандарте XMI определены два метода создания XML-документа:

1) метод расширения пакетов;

2) метод включения объектов.

Препроцессор, соответствующий требованиям настоящего стандарта, должен использовать метод расширения пакетов (см. 5.3.3 и 6 в XMI 1.2).

4.1 Соответствие препроцессора

Соответствующий требованиям препроцессор должен иметь на входе данные, по крайней мере, одного из следующих видов:

- исходное представление, соответствующее ИСО 10303-11;
- исходное представление, соответствующее XML-форме представления схем согласно ИСО 10303-28.

Соответствующий требованиям препроцессор должен принимать на входе метод для определения следующих параметров:

- опции, выбранной для отображения простых типов данных языка EXPRESS (см. 5.3);
- опции включения формулировок на языке OCL в тех случаях, когда это задано в отображении;
- опции, выбранной для обеспечения возможности навигации подклассов AssociationEnd языка UML, получившихся в результате отображения явных атрибутов языка EXPRESS (см. 5.7).

Соответствующий требованиям препроцессор должен генерировать обменный XML-документ, поддерживающий один или несколько XMI-совместимых форматов UML, основываясь на отображениях, определенных в настоящем стандарте.

Примечание — На рисунке 3 показаны стандарты, используемые в препроцессоре, соответствующем требованиям настоящего стандарта.

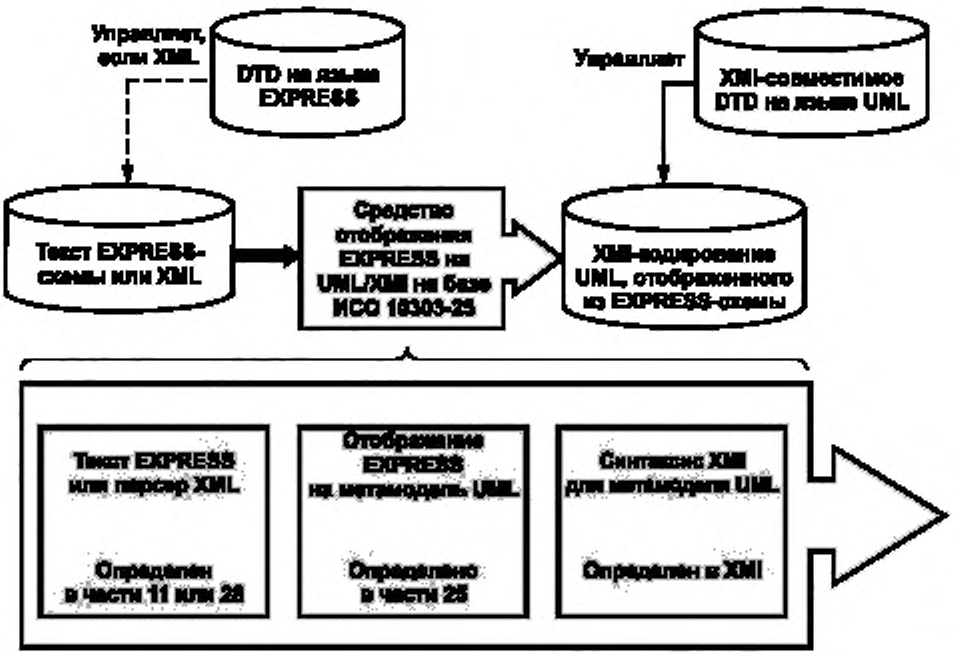

Рисунок 3 — Стандарты, используемые в реализации, соответствующей требованиям настоящего стандарта

4.2 Соответствие обменного документа

Обменный XML-документ будет соответствовать требованиям настоящего стандарта, если выполняются следующие три условия:

- XML-документ соответствует спецификации XMI;
- XML-документ основывается на допустимой контекстной EXPRESS-схеме;
- EXPRESS-схема отображается на метамодель UML, соответствующую настоящему стандарту и представленную в XMI-форме.

5 Отображение конструкций языка EXPRESS на метамодель обмена UML

В данном разделе определены отображения конструкций языка EXPRESS на метамодель обмена UML (см. UML 1.4, раздел 5) при условии, что отображаемое понятие языка EXPRESS определено и соответствующие понятия языка UML идентифицированы. Совокупность понятий языка UML, получившихся в результате отображения конкретной совокупности понятий языка EXPRESS, называется набором объявлений UML [UML declaration set (UDS)]. В качестве входных данных требуется допустимая EXPRESS-схема.

Примечание — На рисунке 4 представлена EXPRESS-G диаграмма частей метамодели обмена UML. Каждый класс UML представлен как объектный тип данных языка EXPRESS. Каждое UML-обобщение представлено как взаимосвязь подтипов языка EXPRESS. Атрибуты языка UML, кроме тех, которые относятся к наименованиям, опущены для краткости. Каждая UML-связь представлена как явный атрибут языка EXPRESS без включения инверсивного атрибута для представления второго конца UML-связи. Полная EXPRESS-схема, на которой основан данный рисунок, приведена в приложении C, а соответствующие части метамодели обмена UML приведены в приложении D.

5.1 Неотображаемые понятия языка EXPRESS

Отображение следующих понятий языка EXPRESS на UML не определено в настоящем стандарте, поэтому их включать в UDS не требуется:

- объявления RULE;

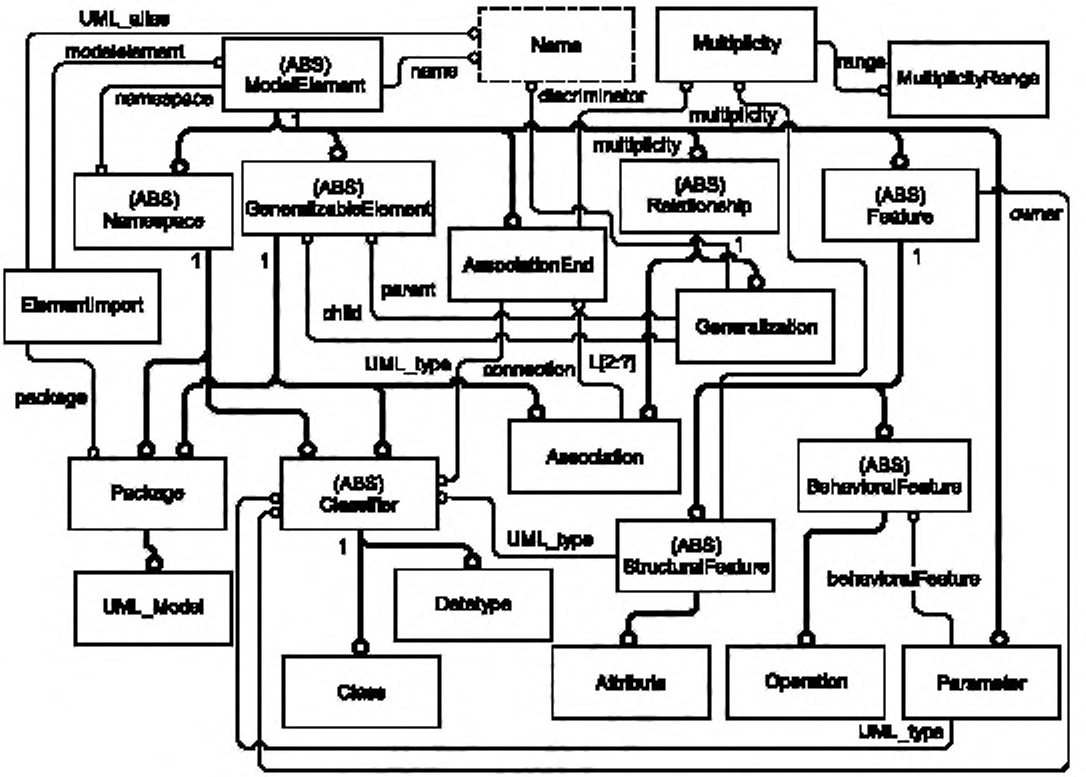

Рисунок 4 — EXPRESS-G диаграмма частей метамодели обмена UML

- правила области определения в объявлениях ENTITY или TYPE;
- правила UNIQUE в объявлениях ENTITY;
- объявления SUPERTYPE, кроме ABSTRACT SUPERTYPE и полных ограничений ONEOF;
- объявления FUNCTION;
- объявления PROCEDURE;
- объявления CONSTANT;
- факультативные ограничения OPTIONAL для атрибутов, областями определения которых являются агрегированные типы данных;
- явные атрибуты, переобъявленные как производные атрибуты;
- комментарии.

Примечания

1 Большинство неотображаемых конструкций являются «правилами» (локальными правилами, глобальными правилами и некоторыми ограничениями на подтипы) и конструкциями, которые могут использоваться только в правилах. Хотя язык описания объектных ограничений (OCL), разработанный в OMG, предназначен для аналогичной цели в UML-моделях, лишь немногие понятия языка EXPRESS, используемые в правилах, имеют точные эквиваленты в OCL.

2 Конструкциями, не отображаемыми на UML, являются в основном конструкции, которые не представляются на EXPRESS-G диаграммах.

5.2 Общие требования к отображению

Спецификация UML 1.4 определяет два механизма, в которых присутствуют объявления: UML-модель и UML-пакет. UML-модель «собирает представление о физической системе»; она «определяется своим назначением и уровнем абстракции (например, модель анализа, модель проектирования, модель

реализации)» и «является полным описанием физической системы». UML-пакет используется «в рамках модели для группировки объектов ModelElement».

EXPRESS — это язык описания данных. В языке EXPRESS схема определяет область определения, а также раздел, в котором присутствуют объявления. Для целей настоящего стандарта должна быть выбрана одна схема для представления контекста реализации. Что касается спецификации данных, то совокупность схем, находящихся в области видимости для данной контекстной схемы, определяет полный набор доступных типов данных.

Основываясь на ключевых понятиях UML, EXPRESS и настоящего стандарта, можно сформулировать следующие требования к формированию групп и разделов:

- в UDS должна присутствовать одна UML-модель, определяющая вид спецификации данных системы, определенной контекстной схемой;

- EXPRESS-схемы должны отображаться на UML-пакеты в рамках данной UML-модели в соответствии с 5.4.

У любого XML-элемента в UDS, имеющего XML-атрибут с именем "visibility" (видимость), представляющий в языке UMLконцепцию видимости объекта, значением данного XML-атрибута должно быть «public» (общедоступный).

5.3 Отображения для простых типов данных языка EXPRESS

Представления на языке UML простых типов данных языка EXPRESS должны присутствовать в UDS в рамках того же пространства имен UML, что и UML-представление объявлений EXPRESS-схемы.

Примечание — Простые типы данных языка EXPRESS не объявляются в схеме, а являются глобальными.

Настоящий стандарт определяет два способа отображения простых типов данных языка EXPRESS. Все простые типы данных языка EXPRESS для конкретного отображения должны отображаться с использованием одного и того же способа.

Первый способ заключается в том, что все простые типы данных языка EXPRESS должны отображаться на классы UML. Именем класса UML должно быть имя простого типа данных языка EXPRESS, начинающееся с прописной буквы, за которой следуют строчные буквы. Класс UML должен иметь значение стереотипа UML "<<primitive>>". У всех концов связей AssociationEnd, связанных с данным классом UML, значением атрибута "navigability" (возможность навигации) должно быть "true" (истина).

Второй способ заключается в том, что простые типы данных языка EXPRESS должны отображаться в соответствии со следующими правилами:

- тип данных INTEGER языка EXPRESS должен отображаться на тип данных Integer языка UML;

- тип данных STRING языка EXPRESS должен отображаться на тип данных String языка UML;

- тип данных BOOLEAN языка EXPRESS должен отображаться на тип данных Boolean языка UML;

- тип данных LOGICAL языка EXPRESS должен отображаться на тип данных Enumeration языка UML с именем "Logical", атрибуты EnumerationLiteral которого должны иметь значение "false", "true" или "unknown". При этом должен применяться стереотип UML "<< enumeration>>";

- тип данных BINARY языка EXPRESS должен отображаться на класс DataType языка UML с именем "Binary".

Примечание — Ни UML, ни MOF не имеют встроенного типа данных, соответствующего типу данных BINARY. В реализациях на основе CORBA может использоваться отображение на встроенный тип данных Octet;

- типы данных NUMBER и REAL языка EXPRESS должны отображаться на класс DataType языка UML с именем "Double". Область определения этого типа данных должна совпадать с областью определения типа данных MOF 1.4 Primitive DataType с именем "Double".

Примечание — UML не имеет встроенного типа данных, соответствующего типам данных REAL или NUMBER, поэтому используется тип данных MOF Double.

5.4 Отображения для объявлений EXPRESS-схем и спецификаций интерфейсов

Конкретная EXPRESS-схема является контекстом для некоторого отображения и называется контекстной схемой. В любом конкретном отображении необходимо рассматривать только данную контекстную схему и EXPRESS-схемы, определенные в ее области видимости с помощью спецификаций интерфейсов языка EXPRESS. Для каждого объявления EXPRESS-схемы, видимой для данной контекстной схемы и видящей данную контекстную схему, в UDS должен присутствовать UML-пакет. Атрибут UML

ModelElement.name UML-пакета должен иметь имя EXPRESS-схемы, начинающееся с прописной буквы, за которой следуют строчные буквы.

Ниже в данном разделе понятие «в контексте отображения» относится только к тем конструкциям языка EXPRESS, которые прямо или косвенно имеют интерфейсы с данной контекстной схемой.

Для каждого оператора USE FROM или REFERENCE FROM языка EXPRESS из основной схемы в UDS должен присутствовать объект UML ElementImport. Значением атрибута UML ElementImport.importedElement должна быть ссылка на UML-пакет, представляющий схему, имеющую интерфейс с данной схемой. Значением атрибута UML ElementImport.package должна быть ссылка на UML-пакет, представляющий соответствующую EXPRESS-схему.

Для каждой отображенной конструкции языка EXPRESS, не являющейся частью схемы, но указанной в операторах USE FROM или REFERENCE FROM языка EXPRESS, в UDS должен присутствовать объект UML ElementImport. Значением атрибута UML ElementImport.importedElement должна быть ссылка на конструкцию языка UML, представляющую связанную через интерфейс конструкцию языка EXPRESS. Значением атрибута UML ElementImport.package должна быть ссылка на UML-пакет, представляющий соответствующую EXPRESS-схему. Если конструкция языка EXPRESS переименовывается с использованием ключевого слова AS языка EXPRESS, то значением атрибута UML ElementImport.alias должно быть новое имя, указанное после ключевого слова AS языка EXPRESS, начинающееся с прописной буквы, за которой следуют строчные буквы.

Для всех объектов UML ElementImport, присутствующих в UDS, значением атрибута ElementImport. visibility должно быть "public".

Конструкции языка UML, представляющие конструкции языка EXPRESS, прямо или косвенно связанные через интерфейс с данной контекстной EXPRESS-схемой, должны присутствовать в UDS в UML-пакете, представляющем EXPRESS-схему, в рамках которой они объявлены.

Кроме конструкций языка UML, представляющих конструкции языка EXPRESS, видимые в данной контекстной схеме, в UDS могут быть включены конструкции языка UML, представляющие конструкции языка EXPRESS, невидимые в данной контекстной схеме. Если такие конструкции имеются, то они должны отображаться в соответствии с настоящим стандартом.

Для каждой конструкции языка UML из UDS, соответствующая UML-ассоциация, связывающая объект UML ModelElement с его пространством имен (объектом UML Namespace), должна присутствовать в UDS. В данной UML-ассоциации значением атрибута ModelElement.namespace должна быть ссылка на UML-пакет, представляющий EXPRESS-схему, в которой объявлен данный элемент, а атрибут Namespace.ownedElement должен содержать ссылку на конструкцию языка UML, представляющую конструкцию языка EXPRESS, объявленную в данной схеме.

5.5 Отображения для объектных типов данных языка EXPRESS

Для каждого объектного типа данных языка EXPRESS в контексте отображения в UDS должен присутствовать класс UML. Значением атрибута UML ModelElement.name данного класса UML должно быть наименование объектного типа данных языка EXPRESS, который он представляет, начинающееся с прописной буквы, за которой следуют строчные буквы. Если объектный тип данных языка EXPRESS объявлен как ABSTRACT SUPERTYPE, то значением атрибута UML GeneralizableElement.isAbstract класса UML должно быть "true", в противном случае — "false". У всех классов UML, представляющих объектные типы данных языка EXPRESS, значением атрибутов UML GeneralizableElement.isLeaf и .isRoot должно быть "false".

5.6 Отображения для объявлений подтипов данных языка EXPRESS

Для каждого объявления SUBTYPE в объектном типе данных языка EXPRESS в контексте отображения, в UDS должно присутствовать UML-обобщение. Значением атрибута Generalization.discriminator должна быть пустая строка.

UML-ассоциация между UML-обобщением и классом UML, представляющим подтип данных языка EXPRESS, должна присутствовать в UDS. Объект UML AssociationEnd с именем "child", должен ссылаться на класс UML, представляющий подтип данных языка EXPRESS, а объект UML AssociationEnd с именем "generalization (обобщение)" должен быть ссылкой на класс UML, представляющий супертип языка EXPRESS.

UML-ассоциация между UML-обобщением и классом UML, представляющим супертип данных языка EXPRESS, должна присутствовать в UDS. Объект UML AssociationEnd с именем "parent (предок)", должен ссылаться на класс UML, представляющий подтип данных языка EXPRESS, а объект UML AssociationEnd

с именем "specialization (специализация)" должен быть ссылкой на класс UML, представляющий подтип языка EXPRESS.

Для каждого объектного типа данных языка EXPRESS, видимого в контексте отображения, который имеет видимые в контексте отображения подтипы и в котором не заданы ограничения супертипа, в UDS должно присутствовать ограничение UML "overlapping (перекрытие)". Для каждого объектного типа данных языка EXPRESS, являющегося подтипом объектного типа данных языка EXPRESS, в котором не заданы ограничения супертипа, UML-обобщение, представляющее взаимосвязь данного подтипа языка EXPRESS, должно быть включено в ограничение UML "overlapping", созданное для данного супертипа языка EXPRESS.

Примечание — Ограничение перекрытия в языке UML эквивалентно оператору ANDOR языка EXPRESS. Это необходимо, так как ограничение непересечения в UML (эквивалентное ONEOF в EXPRESS) используется по умолчанию.

Для каждого объектного типа данных языка EXPRESS, видимого в контексте отображения, который имеет видимые в контексте отображения подтипы, каждый из которых присутствует в одном ограничении SUPERTYPE ONEOF, в UDS должно присутствовать ограничение UML "disjoint (непересечение)". Для каждого объектного типа данных языка EXPRESS, являющегося подтипом объектного типа данных языка EXPRESS, присутствующим в ограничении SUPERTYPE ONEOF, UML-обобщение, представляющее взаимосвязь данного подтипа языка EXPRESS, должно быть включено в ограничение UML "disjoint", созданное для ограничения SUPERTYPE ONEOF языка EXPRESS.

5.7 Отображения для явных и инверсных атрибутов языка EXPRESS

Для каждого явного атрибута объектного типа данных языка EXPRESS в контексте отображения в UDS должны присутствовать атрибут UML либо UML-ассоциация. Значением параметра ModelElement.name атрибута UML или UML-связи должно быть имя атрибута языка EXPRESS, представленное строчными буквами.

Если атрибут языка EXPRESS является повторным объявлением унаследованного атрибута, то его не надо повторять в отображении подтипа. В случае если повторное объявление является конкретизацией объектного типа данных в графе супертипов / подтипов, то в UDS должно присутствовать выражение на языке OCL. Атрибут UML Constraint.name должен иметь значение "redeclared- (повторно объявленный)" с добавлением после дефиса наименования атрибута языка EXPRESS, представленного строчными буквами. Атрибут UML Constraint.body должен иметь следующий вид: context <classname> inv <constraintname> : self . <attributename> -> forAll (oclIsTypeOf (<subclassname>)),

где:

- <classname> — имя класса UML, представляющего объектный тип данных языка EXPRESS, в котором определено данное повторное объявление;
- <constraintname> — атрибут UML Constraint.name;
- <attributename> — наименование повторно объявленного атрибута языка EXPRESS, представленное строчными буквами;
- <subclassname> — наименование класса UML, представляющего область определения повторно объявленного атрибута.

Для атрибута UML, представляющего явный атрибут языка EXPRESS, значением параметра UML Feature.owner должна быть ссылка на класс UML, представляющий объектный тип данных языка EXPRESS. Для атрибута UML начальное значение не устанавливают.

Для UML-ассоциации, представляющей явный атрибут языка EXPRESS, значение атрибута "navigability" у объектов UML AssociationEnd должно задаваться следующим образом:

- если ни один инверсный атрибут языка EXPRESS не связан с явным атрибутом, то значением атрибута "navigability" объекта UML AssociationEnd, связанного с классом UML представляющим объектный тип данных языка EXPRESS, должно быть "false";
- если инверсный атрибут языка EXPRESS связан с явным атрибутом, то значением атрибута «navigability» объекта UML AssociationEnd, связанного с классом UML, представляющим объектный тип данных языка EXPRESS, может быть "true" или "false", и такое же значение должно быть задано для всех инверсных атрибутов в контексте отображения;
- для объекта UML AssociationEnd, связанного с классом UML-классификатором, представляющим область определения явного атрибута языка EXPRESS, значением атрибута "navigability" должно быть "true".

Если явный атрибут языка EXPRESS объявлен как факультативный и он представлен атрибутом UML, то кратность данного атрибута UML должна иметь значение "0 ..1".

Если явный атрибут языка EXPRESS объявлен как факультативный и он представлен UML-ассоциацией, то значением нижней границы кратности объекта UML AssociationEnd должен быть "0" на конце, относящемся к UML-классификатору, представляющему область определения данного явного атрибута.

Остальные отображения явных атрибутов языка EXPRESS зависят от области определения атрибута и определены ниже.

5.7.1 Области определения простых типов данных

Для каждого явного атрибута языка EXPRESS, областью определения которого является простой или перечисляемый тип данных языка EXPRESS, в UDS должен присутствовать атрибут UML с единственным значением. Параметр UML StructuralFeature.type данного атрибута UML должен ссылаться на представление данной области определения на языке UML (см. 5.3 и 5.9.1).

5.7.2 Области определения именованных типов данных и типов данных SET и LIST OF UNIQUE, базирующихся на именованном типе данных

UML-ассоциация и два относящихся к ней объекта UML AssociationEnd должны присутствовать в UDS для каждого явного атрибута языка EXPRESS, областью определения которого является один из следующих типов данных:

- именованный тип данных;
- тип данных SET, базовым типом которого является именованный тип данных;
- тип данных LIST OF UNIQUE, базовым типом которого является именованный тип данных.

Один объект UML AssociationEnd должен ссылаться на класс UML, представляющий именованный тип данных языка EXPRESS, а значением его атрибута UML ModelElement.name должно быть наименование явного атрибута языка EXPRESS, представленное строчными буквами. Если областью определения является именованный тип данных, то кратность данного объекта UML AssociationEnd должна иметь значение "1..1", если данный явный атрибут является обязательным, и "0..1", если он факультативный. Если областью определения является тип данных SET или LIST OF UNIQUE, то кратность объекта UML Association должна иметь значение, отражающее мощность множества типа данных SET или LIST OF UNIQUE, за исключением случая, когда явный атрибут является факультативным. В данном случае значением нижней границы должен быть "0". Если типом данных LIST OF UNIQUE является агрегированный тип данных языка EXPRESS, то у объекта UML AssociationEnd, ссылающегося на представление на языке UML именованного типа данных языка EXPRESS, значением атрибута AssociationEnd.ordering должно быть "ordered (упорядоченный)".

Другой объект UML AssociationEnd должен ссылаться на класс UML, представляющий объектный тип данных языка EXPRESS, с которым явный атрибут языка EXPRESS был объявлен. Кратность данного объекта UML AssociationEnd должна быть равна "0..*", за исключением случая, когда явный атрибут языка EXPRESS имеет соответствующий инверсный атрибут. В данном случае значение кратности должно отражать границы инверсного атрибута. Если атрибут языка EXPRESS имеет соответствующий инверсный атрибут, то значением атрибута UML ModelElement.name данного объекта UML AssociationEnd должно быть имя инверсного атрибута языка EXPRESS, представленное строчными буквами. Если соответствующий инверсный атрибут языка EXPRESS не определен, то значением атрибута UML ModelElement.name может быть имя явного атрибута языка EXPRESS, представленное строчными буквами, с префиксом в виде строки 'reverse-of-', либо данный атрибут может быть опущен.

5.7.3 Области определения типов данных BAG и LIST, базирующихся на именованном типе данных

Для каждого явного атрибута языка EXPRESS, областью определения которого является тип данных BAG или неуникальный тип данных LIST, базовым типом которого является именованный тип данных языка EXPRESS, в UDS должен присутствовать многозначный атрибут UML. Параметр UML StructuralFeature.type данного атрибута UML должен ссылаться на класс UML, представляющий именованный тип данных языка EXPRESS. Значение кратности объекта UML StructuralFeature должно отражать мощность множества типа данных BAG или неуникального LIST, за исключением случая, когда явный

атрибут является факультативным. В данном случае значением нижней границы должен быть "0". Если тип данных LIST не является уникальным, то значением атрибута UML StructuralFeature.ordering должно быть "ordered".

5.7.4 Области определения других агрегированных типов данных

Для каждого явного атрибута языка EXPRESS, областью определения которого является агрегированный тип данных, не относящийся к типу данных SET, BAG или LIST, базирующемуся на именованном типе данных, в UDS должна присутствовать UML-ассоциация, представляющая явный атрибут языка EXPRESS. UML-ассоциация должна иметь два объекта UML AssociationEnd: один, ссылающийся на представление на языке UML объектного типа данных языка EXPRESS с агрегированным типом данных в качестве области определения атрибута, и второй, ссылающийся на представление на языке UML типа данных, являющегося базовым для агрегированного типа данных языка EXPRESS (см. 5.10). Значением кратности UML-ассоциации должно быть "1..1" на конце UML AssociationEnd, связанном с классом UML, представляющим объектный тип данных языка EXPRESS, за исключением случая, когда явный атрибут является факультативным. В данном случае значением нижней границы должен быть "0". Значением кратности UML-ассоциации должно быть "0..*" на конце UML AssociationEnd, связанном с областью определения явного атрибута.

5.8 Отображения для вычисляемых атрибутов языка EXPRESS

Для каждого вычисляемого атрибута объектного типа данных языка EXPRESS в контексте отображения в UDS должна присутствовать UML-операция класса UML, представляющего данный объектный тип данных. Значением атрибута ModelElement.name данной UML-операции должно быть имя вычисляемого атрибута языка EXPRESS, представленное строчными буквами. Значением атрибута UML BehavioralFeature.isQuery должно быть 'true'. Единственный UML-параметр, связанный с UML-операцией, должен присутствовать в UDS, представляя тип области определения вычисляемого атрибута языка EXPRESS. Значением атрибута UML Parameter.kind должно быть 'return'. Связь атрибута UML Parameter.parameter должна ссылаться на UML-операцию, представляющую вычисляемый атрибут языка EXPRESS. Связь атрибута UML Parameter.type должна ссылаться на класс UML или тип данных UML, представляющий область определения вычисляемого атрибута языка EXPRESS.

5.9 Отображения для определенных типов данных языка EXPRESS

Каждый из трех видов определенного типа данных языка EXPRESS отображается на разные конструкции языка UML. Перечисляемые типы данных языка EXPRESS имеют прямое соответствие с UML-перечислениями. Однако другие определенные типы данных языка EXPRESS отображаются на классы UML.

5.9.1 Отображения для перечисляемых типов данных языка EXPRESS

Для каждого перечисляемого типа данных языка EXPRESS в контексте отображения в UDS должно присутствовать UML-перечисление. Значением атрибута ModelElement.name данного UML-перечисления должно быть наименование перечисляемого типа данных языка EXPRESS, начинающееся с прописной буквы, за которой следуют строчные буквы. Значениями объектов UML EnumerationLiteral должны быть наименования элементов перечисления языка EXPRESS, представленные строчными буквами, в том же порядке, в котором они определены в объявлении перечисляемого типа данных языка EXPRESS. UML-перечисление должно иметь UML-стереотип со значением "<<enumeration>>".

5.9.2 Отображения для выбираемых типов данных языка EXPRESS

Для каждого выбираемого типа данных языка EXPRESS в контексте преобразования в UDS должен присутствовать класс UML. Значением атрибута ModelElement.name должно быть наименование выбираемого типа данных языка EXPRESS, начинающееся с прописной буквы, за которой следуют строчные буквы. Для каждого именованного типа данных языка EXPRESS из списка выбора, подлежащего отображению, в UDS должна присутствовать UML-ассоциация агрегации, не являющейся композитной агрегацией. Значением атрибута UML Association.name должно быть "selection_of". UML-ассоциация должна иметь точно два конца связи UML AssociationEnd. Значением атрибута UML AssociationEnd.aggregation должно быть "aggregate" на конце, указывающем на класс UML, представляющий выбираемый тип данных языка EXPRESS, и "none" на конце, указывающем на класс UML, представляющий элемент из списка выбора

языка EXPRESS. Ограничение XOR языка UML должно применяться к множеству UML-ассоциаций, связывающих класс UML, представляющий выбираемый тип данных языка EXPRESS, с классами UML, представляющими элементы из списка выбора языка EXPRESS.

5.9.3 Отображения для неконструкционных определенных типов данных языка EXPRESS

Для каждого определенного типа данных языка EXPRESS, не являющегося конструкционным типом данных, в UDS должен присутствовать класс UML, представляющий определенный тип данных. Значением атрибута ModelElement.name класса UML, представляющего определенный тип данных языка EXPRESS, должно быть наименование определенного типа данных языка EXPRESS, начинающееся с прописной буквы, за которой следуют строчные буквы. В UDS должно присутствовать UML-обобщение, связывающее представление на языке UML определенного типа данных с его базовым типом. Значением атрибута Generalization.discriminator должна быть пустая строка. Должна существовать UML-ассоциация, связывающая объект GeneralizableElement, представляющий определенный тип данных, с UML-обобщением. При этом роль UML-обобщения должна быть определена как "child (потомок)", а конец, соответствующий объекту UML GeneralizableElement, должен быть определен как "specialization (специализация)". Должна существовать UML-ассоциация, связывающая объект UML GeneralizableElement, представляющий базовый тип данных, с UML-обобщением. При этом роль UML-обобщения должна быть определена как "parent (предок)", а конец, соответствующий объекту UML GeneralizableElement, должен быть определен как "generalization (обобщение)". Для определения предка в UML-обобщении следует применять одно из следующих правил:

- если базовым типом является другой определенный тип данных языка EXPRESS, то класс UML, представляющий определенный тип данных языка EXPRESS, являющийся базовым типом, должен быть предком в UML-обобщении;
- если базовым типом является простой тип данных языка EXPRESS, то представление на языке UML этого простого типа данных языка EXPRESS должно быть предком в UML-обобщении;
- если базовым типом является агрегированный тип данных языка EXPRESS, то класс UML, представляющий агрегированный тип данных языка EXPRESS в соответствии с 5.10, должен быть предком в UML-обобщении.

5.10 Отображения для агрегированных типов данных языка EXPRESS

Язык EXPRESS поддерживает агрегированные типы данных, для которых не существует прямого эквивалента в языке UML. По этой причине, даже если агрегированные типы данных языка EXPRESS имеют сходные структуры, при отображении на язык UML результатом не всегда будут такие же структуры. Каждое использование агрегированных типов данных языка EXPRESS трактуется в настоящем стандарте следующим образом:

- отображение для использования агрегированного типа данных языка EXPRESS в качестве базового типа для определенного типа данных, установлено в 5.9.3 и в последующей части настоящего подраздела;
- отображение для использования агрегированного типа данных языка EXPRESS в качестве базового типа для другого агрегированного типа данных, установлено в последующей части настоящего подраздела;
- отображение для использования агрегированного типа данных языка EXPRESS в качестве объявленного типа данных для локальной переменной или параметра функции не установлено в настоящем стандарте, так как оператор FUNCTION языка EXPRESS не отображается на язык UML;
- отображение для использования агрегированного типа данных языка EXPRESS в качестве области определения явного атрибута трактуется по-разному, в зависимости от деталей его использования. Отображение для явных атрибутов языка EXPRESS с областями определения именованных типов данных SET или LIST OF UNIQUE установлено в 5.7.2. Отображение для явных атрибутов языка EXPRESS с областями определения именованных типов данных BAG или LIST OF установлено в 5.7.3. В этих случаях никакой класс UML сам по себе не представляет агрегированный тип данных языка EXPRESS. Для остальных случаев отображение установлено в последующей части настоящего подраздела.

Примечание — Взаимосвязи в языке EXPRESS определяются с помощью атрибутов, а не с помощью специальной конструкции, каковой в языке UML является UML-ассоциация. В языке EXPRESS множественные взаимосвязи определяются базовым типом атрибута языка EXPRESS, являющимся агрегированным типом данных. В языке UML связи между объектами UML-классификаторами являются математическими множествами, поддерживающими множественную взаимосвязь, но не допускающими дублирования (см. D2). По этим причи-

нам, единственным случаем, когда структура агрегированного типа данных языка EXPRESS отображается на UML-ассоциации, является использование явных атрибутов языка EXPRESS, областью определения которых является тип данных SET или LIST OF UNIQUE, базовым типом которого является именованный тип данных языка EXPRESS (см. 5.7.2). Другие агрегированные типы данных языка EXPRESS не отображаются на UML-ассоциации, поэтому тот факт, что они могут представлять взаимосвязь в EXPRESS-схеме, является менее очевидным в результирующем отображении на UML.

Для всех применений агрегированных типов данных языка EXPRESS, кроме определенных в 5.7.2 и 5.7.3, в UDS должен присутствовать один или несколько классов UML, представляющих агрегированный тип данных языка EXPRESS. В последующей части настоящего подраздела подробно описаны данные классы и другие имеющие к ним отношение конструкции языка UML, которые должны присутствовать в UDS.

5.10.1 Базовые типы агрегированных типов данных, являющиеся агрегированными типами данных

Если базовым типом агрегированного типа данных языка EXPRESS является агрегированный тип данных (или любое число последующих вложений агрегированных типов данных языка EXPRESS), то он должен быть представлен двумя или более классами UML.

Класс UML, представляющий самый последний из вложенных агрегированных типов данных языка EXPRESS, для которого базовым типом не является другой агрегированный тип данных, должен присутствовать в UDS в соответствии с 5.10.2.

Класс UML, представляющий каждый уровень вложения агрегированных типов данных языка EXPRESS, должен присутствовать в UDS. Значение атрибута UML ModelElement.name каждого из этих классов UML должно быть конкатенацией следующих составляющих:

1) наименования вида агрегированного типа данных языка EXPRESS (Set, Bag, List или Array), начинающегося с прописной буквы, за которой следуют строчные буквы;

2) строки, представляющей целое число, обозначающее нижнюю границу, за которой следует строка "-", за которой, в свою очередь, следует строка, представляющая целое число, обозначающее верхнюю границу, или строка "unbounded (неограниченный)" в тех случаях, когда взаимосвязь между агрегированным типом данных и его базовым типом представлена многозначным атрибутом UML;

3) строки "-of-";

4) значения атрибута UML ModelElement.name, представленного всеми строчными буквами, конструкции языка UML, представляющей базовый тип агрегированного типа данных языка EXPRESS.

Для каждой пары классов UML, представляющих агрегированный тип данных языка EXPRESS (или отдельный уровень вложения в многократно вложенном агрегированном типе данных), в UDS должна присутствовать UML-ассоциация для агрегированного типа данных или многозначный атрибут UML в соответствии с 5.10.3.

5.10.2 Базовые типы агрегированных типов данных, не являющиеся агрегированными типами данных

Если базовым типом агрегированного типа данных языка EXPRESS является неагрегированный тип данных, то он должен быть представлен одним классом UML. Значение атрибута UML ModelElement.name данного класса UML должно быть конкатенацией следующих составляющих:

1) наименования вида агрегированного типа данных языка EXPRESS (Set, Bag, List или Array), начинающегося с прописной буквы, за которой следуют строчные буквы;

2) строки, представляющей целое число, обозначающее нижнюю границу, за которой следует строка "-", за которой, в свою очередь, следует строка, представляющая целое число, обозначающее верхнюю границу, или строка "unbounded" (неограниченный) в тех случаях, когда взаимосвязь между агрегированным типом данных и его базовым типом представлена многозначным атрибутом UML;

3) строки "-of-";

4) значения атрибута UML ModelElement.name, представленного всеми строчными буквами, конструкции языка UML, представляющей базовый тип агрегированного типа данных языка EXPRESS.

В UDS должна присутствовать UML-ассоциация для агрегированного типа данных или многозначный атрибут UML для того, чтобы связать агрегированный тип данных языка EXPRESS с его базовым типом в соответствии с 5.10.3.

Примечание — Отображения явных атрибутов языка EXPRESS с базовыми типами данных SET, LIST и BAG, базирующимися на именованных типах данных, определены в 5.7.2 и 5.7.3.

5.10.3 Связь агрегированных типов данных с их базовыми типами

Для каждого агрегированного типа данных языка EXPRESS, т. е. BAG, ARRAY или неуникального LIST, в классе UML должен присутствовать многозначный атрибут UML, представляющий этот агрегированный тип данных языка EXPRESS. Значением параметра UML Attribute.name должно быть "elements (элементы)". Атрибут UML должен быть задан как ссылка на Class, представляющий базовый тип агрегированного типа данных языка EXPRESS. Для типов данных BAG или LIST языка EXPRESS значение кратности атрибута UML должно отражать мощность множества агрегированного типа данных языка EXPRESS. Для типа данных ARRAY языка EXPRESS значение кратности атрибута UML должно отражать число элементов в агрегированном типе данных языка EXPRESS. Если атрибут UML представляет тип данных ARRAY или неуникальный LIST языка EXPRESS, то значением параметра UML StructuralFeature.ordering должно быть "ordered (упорядоченный)".

Для каждого агрегированного типа данных SET или LIST OF UNIQUE языка EXPRESS в UDS должна присутствовать совместно используемая UML-ассоциация для агрегированного типа данных, атрибут UML Association.name которой должен иметь значение "aggregation_of". UML-ассоциация должна иметь точно два конца связи UML AssociationEnd. Атрибут UML AssociationEnd.aggregation, относящийся к классу UML, представляющему агрегированный тип данных языка EXPRESS, должен иметь значение "aggregate". Атрибут UML AssociationEnd.aggregation, относящийся к классу UML, представляющему базовый тип агрегированного типа данных языка EXPRESS, должен иметь значение "none". Значение кратности объекта UML AssociationEnd должно отображать мощность множества агрегированного типа данных языка EXPRESS. В случае если агрегированным типом данных языка EXPRESS является ARRAY или LIST OF UNIQUE, то у объекта UML AssociationEnd, относящегося к представлению на языке UML базового типа агрегированного типа данных языка EXPRESS, атрибут UML AssociationEnd.ordering должен иметь значение "ordered".

Приложение А
(обязательное)

Регистрация информационного объекта

Для однозначного обозначения информационного объекта в открытой системе настоящему стандарту присвоен следующий идентификатор объекта:

{ iso standard 10303 part(25) version(1) }

Смысл данного обозначения установлен в ИСО/МЭК 8824-1 и описан в ИСО 10303-1.

Приложение В
(справочное)

Примеры схем, DTD, UML-диаграмм и XMI-файлов

Настоящее приложение содержит примеры EXPRESS-схем, DTD, UML-диаграмм и XMI-файлов.

B.1 Стандартные XMI DTD для UML

OMG опубликовала несколько соответствующих XMI шаблонов DTD для обмена UML-моделями. Примерами этих шаблонов DTD являются:

- шаблон DTD для обмена моделями UML 1.4 и 1.5, документ OMG от 01-02-16;
- шаблон DTD для UML 1.3, на основе XMI 1.0.

Эти шаблоны DTD поддерживают весь язык UML и поэтому содержат много объявлений, не используемых в отображениях, определенных в настоящем стандарте.

В.2 Пример — три связанные схемы

В приложении Е представлен итоговый пример отображения, определенного в настоящем стандарте. Приведенные в приложении Е фрагменты EXPRESS-схем взяты из схемы: **context_schema**, связанной со схемами **s2_schema** и **s3_schema.**

Файл примера с XMI 1.2 содержит представления большинства конструкций, присутствующих в этих трех схемах.

Примечание — Следующие конструкции не включены в пример, так как инструментарий UML, используемый для создания XMI-файла, не поддерживает данные конструкции:

- ONE OF;
- REFERENCE FROM;
- USE FROM одиночный объект и AS для его переименования;
- литералы перечисляемого типа.

```
SCHEMA context_schema;

- - данная схема является контекстной схемой

REFERENCE FROM s2_schema;

USE FROM s3_schema (entity1 AS renentity1);

TYPE type1 = SELECT (entity1,type2);

END_TYPE;

TYPE type2 = ENUMERATION OF (red, white);

END_TYPE;

TYPE type3 = STRING;

END_TYPE;

TYPE type4 = LIST OF STRING;

END_TYPE;

ENTITY entity2 ABSTRACT SUPERTYPE;
    a5 : LIST OF type1;
    a6 : LIST OF UNIQUE entity4;
    a7 : REAL;
DERIVE
    da1 : REAL := 2.4*a7;
END_ENTITY;

ENTITY entity1
    SUPERTYPE OF (ONEOF(entity4, entity5))
    SUBTYPE OF (entity2);
    a11 : SET OF entity10;
END_ENTITY;

ENTITY entity4
    SUBTYPE OF (entity1);
    a2 : entity5;
    a8 : LIST OF STRING;
    a12 : type4;
    a13 : renentity1;
END_ENTITY;

ENTITY entity5
    SUBTYPE OF (entity1);
INVERSE
    ia1 : SET[1:?] OF entity4 FOR a2;
END_ENTITY;

END_SCHEMA;

SCHEMA s2_schema;

- - данная схема получает данные через ссылки из context_schema
```

```
ENTITY entity10;
    a1 : INTEGER;
END_ENTITY;

END_SCHEMA;

SCHEMA s3_schema;

- - часть данной схемы получает данные, используемые в context_schema

ENTITY entity1;
    a7 : SET OF STRING;
END_ENTITY;

ENTITY e99;
    a99 : STRING;
END_ENTITY;

END_SCHEMA;
```

B.3 Пример — сад мистера Смита (mr smiths garden)

Следующий фрагмент на языке EXPRESS является подмножеством схемы, отображение которой на UML дает результат, приведенный на рисунке B.1.

```
SCHEMA mr_smiths_garden;

ENTITY garden;
has_bed : bed;
END_ENTITY;

ENTITY bed;
has_plants : SET[0:?] OF plant;
INVERSE
the_garden : SET[0:?] OF garden FOR has_bed;
END_ENTITY;

ENTITY plant;
name : STRING;
END_ENTITY;

END_SCHEMA;
```

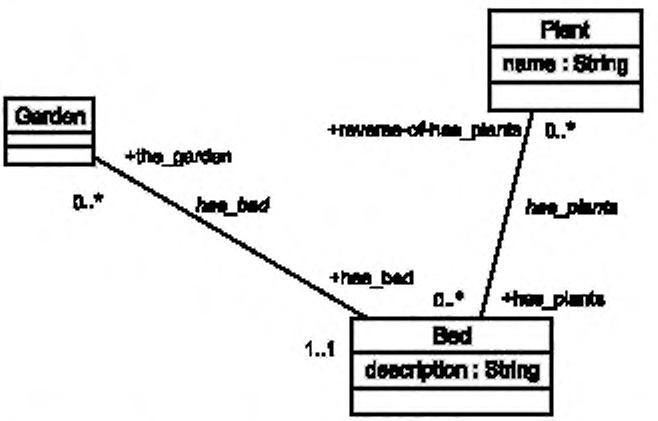

Рисунок B.1 — Диаграмма классов UML, показывающая только объявления из **mr_smiths_garden** в контексте **mr_jones_garden**

Ниже представлен соответствующий файл XMI 1.0, основанный на UML 1.3. Представления на языке UML идентификаторов и типов данных языка EXPRESS выделены полужирным шрифтом для облегчения восприятия. Атрибут XML xmi.id является идентификатором XMI для каждого элемента XML. Атрибут XML xmi.idref представляет ссылку на элемент XML с атрибутом xmi.id, имеющим то же самое значение.

```
<?xml version="1.0" encoding="UTF-8"?>
<XMI xmi.version="1.0">
<XMI.header>
  <XMI.documentation>
     <XMI.exporter>Novosoft UML Library</XMI.exporter>
     <XMI.exporterVersion>0.4.19</XMI.exporterVersion>
   </XMI.documentation>
   <XMI.metamodel xmi.name="UML" xmi.version="1.3"/>
</XMI.header>
<XMI.content>
   <Model_Management.Package xmi.id="xmi.1" xmi.uuid="-63-14-84-52-2d157f:e7cc2f1fc0:-7fff">
    <Foundation.Core.ModelElement.name>Mr_smiths_garden</Foundation.Core.
  ModelElement.name>
     <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
     <Foundation.Core.GeneralizableElement.isRoot xmi.value="false"/>
     <Foundation.Core.GeneralizableElement.isLeaf xmi.value="false"/>
     <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
     <Foundation.Core.Namespace.ownedElement>
        <Foundation.Core.Class xmi.id="xmi.2" xmi.uuid="-63-14-84-52-2d157f:e7cc2f1fc0:-7ffe">
           <Foundation.Core.ModelElement.name>Garden</Foundation.Core.ModelElement. name>
           <Foundation.Core.ModelElement.visibility xmi.value="public"/>
           <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isRoot xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isLeaf xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
           <Foundation.Core.Class.isActive xmi.value="false"/>
           <Foundation.Core.ModelElement.namespace>
              <Foundation.Core.Namespace xmi.idref="xmi.1"/>
           </Foundation.Core.ModelElement.namespace>
        </Foundation.Core.Class>
        <Foundation.Core.Class xmi.id="xmi.3" xmi.uuid="-63-14-84-52-2d157f:e7cc2f1fc0:-7ffc">
           <Foundation.Core.ModelElement.name>Bed</Foundation.Core.ModelElement.name>
           <Foundation.Core.ModelElement.visibility xmi.value="public"/>
           <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isRoot xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isLeaf xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
           <Foundation.Core.Class.isActive xmi.value="false"/>
           <Foundation.Core.ModelElement.namespace>
              <Foundation.Core.Namespace xmi.idref="xmi.1"/>
           </Foundation.Core.ModelElement.namespace>
        </Foundation.Core.Class>
        <Foundation.Core.Class xmi.id="xmi.4" xmi.uuid="-63-14-84-52-2d157f:e7cc2f1fc0:-7ffb">
           <Foundation.Core.ModelElement.name>Plant</Foundation.Core.ModelElement.name>
           <Foundation.Core.ModelElement.visibility xmi.value="public"/>
           <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isRoot xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isLeaf xmi.value="false"/>
           <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
           <Foundation.Core.Class.isActive xmi.value="false"/>
           <Foundation.Core.ModelElement.namespace>
              <Foundation.Core.Namespace xmi.idref="xmi.1"/>
           </Foundation.Core.ModelElement.namespace>
           <Foundation.Core.Classifier.feature>
              <Foundation.Core.Attribute xmi.id="xmi.5">
                 <Foundation.Core.ModelElement.name>name</Foundation.Core.ModelElement.name>
                 <Foundation.Core.ModelElement.visibility xmi.value="public"/>
                 <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
                 <Foundation.Core.Feature.ownerScope xmi.value="instance"/>
                 <Foundation.Core.Feature.owner>
                    <Foundation.Core.Classifier xmi.idref="xmi.4"/>
                 </Foundation.Core.Feature.owner>
```

```
            <Foundation.Core.StructuralFeature.type>
              <Foundation.Core.Classifier xmi.idref="xmi.7"/>
            </Foundation.Core.StructuralFeature.type>
          </Foundation.Core.Attribute>
        </Foundation.Core.Classifier.feature>
      </Foundation.Core.Class>
      <Foundation.Core.Association xmi.id="xmi.8" xmi.uuid="-63-14-84-52-2d157f:e7cc2f1fc0:-7ffa">
        <Foundation.Core.ModelElement.name>has_bed</Foundation.Core.ModelElement.name>
        <Foundation.Core.ModelElement.isSpecification xmi.value= "false"/>
        <Foundation.Core.GeneralizableElement.isRoot xmi.value="false"/>
        <Foundation.Core.GeneralizableElement.isLeaf xmi.value="false"/>
        <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
        <Foundation.Core.ModelElement.namespace>
          <Foundation.Core.Namespace xmi.idref="xmi.1"/>
        </Foundation.Core.ModelElement.namespace>
        <Foundation.Core.Association.connection>
          <Foundation.Core.AssociationEnd xmi.id="xmi.9">
          <Foundation.Core.ModelElement.name>the_garden</Foundation.Core.ModelElement.name>
            <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
            <Foundation.Core.AssociationEnd.isNavigable xmi.value="true"/>
            <Foundation.Core.AssociationEnd.aggregation xmi.value="none"/>
            <Foundation.Core.AssociationEnd.multiplicity>
              <Foundation.Data_Types.Multiplicity xmi.id="xmi.10">
                <Foundation.Data_Types.Multiplicity.range>
                  <Foundation.Data_Types.MultiplicityRange xmi.id="xmi.11">
                    <Foundation.Data_Types.MultiplicityRange.lower>0<
                     /Foundation.Data_Types.MultiplicityRange.lower>
                    <Foundation.Data_Types.MultiplicityRange.upper>-1<
                     /Foundation.Data_Types.MultiplicityRange.upper>
                  </Foundation.Data_Types.MultiplicityRange>
                  </Foundation.Data_Types.Multiplicity.range>
                </Foundation.Data_Types.Multiplicity>
              </Foundation.Core.AssociationEnd.multiplicity>
              <Foundation.Core.AssociationEnd.association>
                <Foundation.Core.Association xmi.idref="xmi.8"/>
              </Foundation.Core.AssociationEnd.association>
              <Foundation.Core.AssociationEnd.type>
                <Foundation.Core.Classifier xmi.idref="xmi.2"/>
              </Foundation.Core.AssociationEnd.type>
            </Foundation.Core.AssociationEnd>
            <Foundation.Core.AssociationEnd xmi.id="xmi.12">
              <Foundation.Core.ModelElement.name>has_bed</Foundation.Core.
               ModelElement.name>
              <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
              <Foundation.Core.AssociationEnd.isNavigable xmi.value="true"/>
              <Foundation.Core.AssociationEnd.aggregation xmi.value="none"/>
              <Foundation.Core.AssociationEnd.multiplicity>
              <Foundation.Data_Types.Multiplicity xmi.id="xmi.13">
                <Foundation.Data_Types.Multiplicity.range>
                  <Foundation.Data_Types.MultiplicityRange xmi.id="xmi.14">
                    <Foundation.Data_Types.MultiplicityRange.lower>1</
                     Foundation.Data.MultiplicityRange.lower>
                    <Foundation.Data_Types.MultiplicityRange.upper>1<
                     /Foundation.Data_Types.MultiplicityRange.upper>
                  </Foundation.Data_Types.MultiplicityRange>
                </Foundation.Data_Types.Multiplicity.range>
              </Foundation.Data_Types.Multiplicity>
            </Foundation.Core.AssociationEnd.multiplicity>
            <Foundation.Core.AssociationEnd.association>
              <Foundation.Core.Association xmi.idref="xmi.8"/>
              </Foundation.Core.AssociationEnd.association>
               <Foundation.Core.AssociationEnd.type>
```

```
        <Foundation.Core.Classifier xmi.idref="xmi.3"/>
      </Foundation.Core.AssociationEnd.type>
    </Foundation.Core.AssociationEnd>
  </Foundation.Core.Association.connection>
</Foundation.Core.Association>
<Foundation.Core.Association xmi.id="xmi.28" xmi.uuid="-63-14-84-52-2d157f:e7cc2f1fc0:-7fea">
  <Foundation.Core.ModelElement.name>has_plants</Foundation.Core.ModelElement.name>
  <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
  <Foundation.Core.GeneralizableElement.isRoot xmi.value="false"/>
  <Foundation.Core.GeneralizableElement.isLeaf xmi.value="false"/>
  <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
  <Foundation.Core.ModelElement.namespace>
    <Foundation.Core.Namespace xmi.idref="xmi.1"/>
  </Foundation.Core.ModelElement.namespace>
  <Foundation.Core.Association.connection>
    <Foundation.Core.AssociationEnd xmi.id="xmi.29">
      <Foundation.Core.ModelElement.name>reverse-of-has_plants<
       /Foundation.Core.ModelElement.name>
      <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
      <Foundation.Core.AssociationEnd.isNavigable xmi.value="true"/>
      <Foundation.Core.AssociationEnd.aggregation xmi.value="none"/>
      <Foundation.Core.AssociationEnd.multiplicity>
        <Foundation.Data_Types.Multiplicity xmi.id="xmi.30">
          <Foundation.Data_Types.Multiplicity.range>
            <Foundation.Data_Types.MultiplicityRange xmi.id="xmi.31">
            <Foundation.Data_Types.MultiplicityRange.lower>0<
             /Foundation.Data_Types.MultiplicityRange.lower>
            <Foundation.Data_Types.MultiplicityRange.upper>-1<
             /Foundation.Data_Types.MultiplicityRange.upper>
          </Foundation.Data_Types.MultiplicityRange>
        </Foundation.Data_Types.Multiplicity.range>
       </Foundation.Data_Types.Multiplicity>
      </Foundation.Core.AssociationEnd.multiplicity>
      <Foundation.Core.AssociationEnd.association>
        <Foundation.Core.Association xmi.idref="xmi.3"/>
      </Foundation.Core.AssociationEnd.association>
      <Foundation.Core.AssociationEnd.type>
        <Foundation.Core.Classifier xmi.idref="xmi.4"/>
      </Foundation.Core.AssociationEnd.type>
    </Foundation.Core.AssociationEnd>
    <Foundation.Core.AssociationEnd xmi.id="xmi.32">
      <Foundation.Core.ModelElement.name>has_plants<
       /Foundation.Core.ModelElement.name>
      <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
      <Foundation.Core.AssociationEnd.isNavigable xmi.value="true"/>
      <Foundation.Core.AssociationEnd.aggregation xmi.value="none"/>
      <Foundation.Core.AssociationEnd.multiplicity>
      <Foundation.Data_Types.Multiplicity xmi.id="xmi.33">
        <Foundation.Data_Types.Multiplicity.range>
          <Foundation.Data_Types.MultiplicityRange xmi.id="xmi.34">
            <Foundation.Data_Types.MultiplicityRange.lower>0<
             /Foundation.Data_Types.MultiplicityRange.lower>
            <Foundation.Data_Types.MultiplicityRange.upper>-1<
             /Foundation.Data_Types.MultiplicityRange.upper>
          </Foundation.Data_Types.MultiplicityRange>
        </Foundation.Data_Types.Multiplicity.range>
      </Foundation.Data_Types.Multiplicity>
    </Foundation.Core.AssociationEnd.multiplicity>
    <Foundation.Core.AssociationEnd.association>
      <Foundation.Core.Association xmi.idref="xmi.28"/>
      </Foundation.Core.AssociationEnd.association>
       <Foundation.Core.AssociationEnd.type>
```

```
                    <Foundation.Core.Classifier xmi.idref="xmi.4"/>
                  </Foundation.Core.AssociationEnd.type>
                </Foundation.Core.AssociationEnd>
              </Foundation.Core.Association.connection>
            </Foundation.Core.Association>
          <Foundation.Core.DataType xmi.id="xmi.7">
            <Foundation.Core.ModelElement.name>String</Foundation.Core.
              ModelElement.name>
            <Foundation.Core.ModelElement.isSpecification xmi.value="false"/>
            <Foundation.Core.GeneralizableElement.isRoot xmi.value="false"/>
            <Foundation.Core.GeneralizableElement.isLeaf xmi.value="false"/>
            <Foundation.Core.GeneralizableElement.isAbstract xmi.value="false"/>
            <Foundation.Core.ModelElement.namespace>
              <Foundation.Core.Namespace xmi.idref="xmi.1"/>
            </Foundation.Core.ModelElement.namespace>
          </Foundation.Core.DataType>
        </Foundation.Core.Namespace.ownedElement>
      </Model_Management.Package>
    </XMI.content>
  </XMI>
```

B.4 Пример отображения EXPRESS на UML для агрегированных типов данных

Отображение на UML приведенного ниже фрагмента на языке EXPRESS, содержащего типы данных SET и LIST OF UNIQUE, базовым типом которых является именованный тип данных, дает результат, приведенный на рисунке B.2.

```
TYPE t = SELECT ( entity1, entity2 );
END_TYPE;

ENTITY entity1;
END_ENTITY;

ENTITY entity2;
a1 : SET[1:4] OF entity1; - - UML maps to Association named a1
a2 : LIST[0:?] OF UNIQUE t; - - maps to UML Association named a2
END_ENTITY;
```

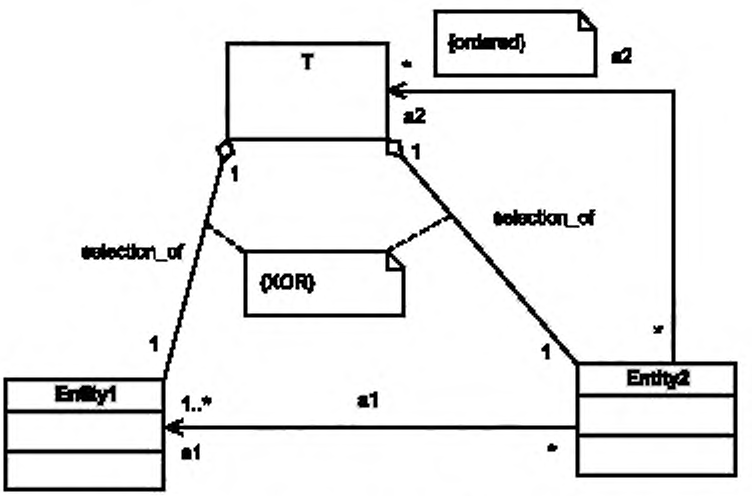

Рисунок B.2 — Пример диаграммы классов UML, на которой показаны только UML-ассоциации, отображенные из SET и LIST OF UNIQUE

Следующий фрагмент на языке EXPRESS, содержащий тип данных LIST, базовым типом данных которого является именованный тип данных, может дать результат в виде UML-диаграммы, показанной на рисунке B.3.

```
TYPE t = SELECT ( entity1, entity2 );
END_TYPE;
```

```
ENTITY entity1;
END_ENTITY;

ENTITY entity2;
a3 : LIST[0:?] OF t; - - maps to UML Attribute named a3
END_ENTITY;
```

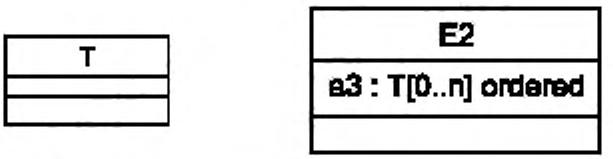

Рисунок В.3 — Пример UML-диаграммы, показывающей атрибут UML, отображенный из типа данных LIST

Приведенный ниже фрагмент на языке EXPRESS, содержащий тип данных LIST, базовым типом которого является именованный тип данных, может дать результат в виде UML-диаграммы, показанной на рисунке В.4.

```
TYPE t = SELECT ( entity1, entity2 );
END_TYPE;

TYPE ag = LIST OF UNIQUE entity1; - - maps to UML Class List-of-entity1
END_TYPE;

ENTITY entity1;
END_ENTITY;

ENTITY entity2;
a4 : ARRAY[1:2] OF ARRAY[1:2] OF t; - - maps to UML Classes
                                    - - Array-of-array-of-t and Array-of-t
a5 : SET OF STRING; - - maps to UML Class Set-of-string
a6 : ARRAY[1:4] OF INTEGER; - - maps to UML Class Array-of-integer
END_ENTITY;
```

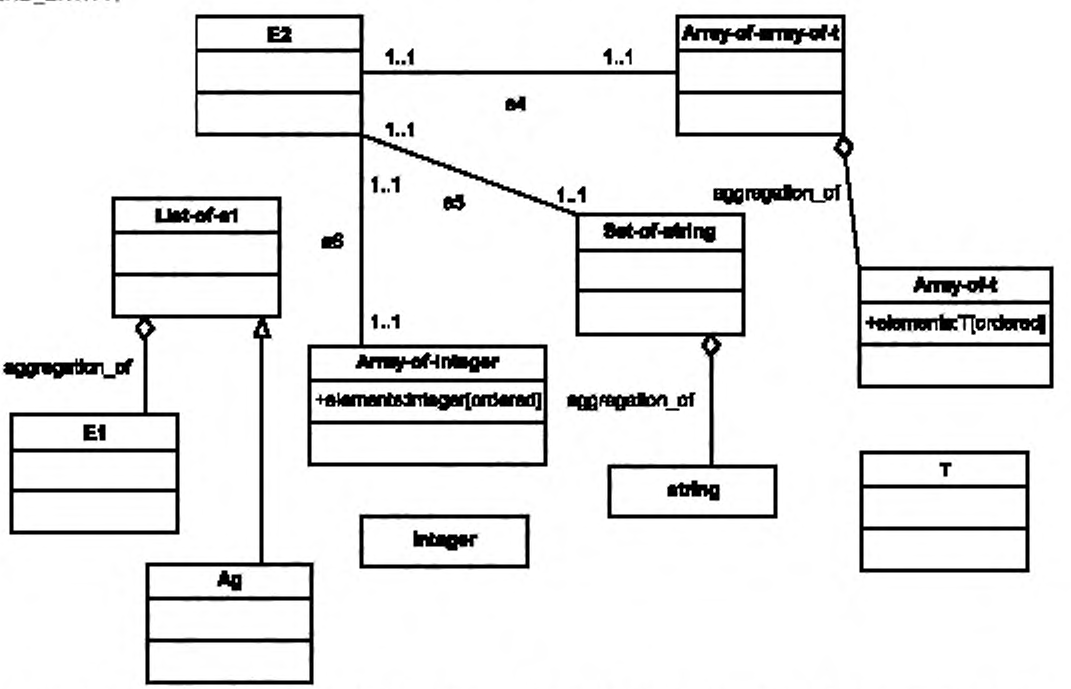

Рисунок В.4 — Пример UML-диаграммы, показывающей классы UML, отображенные из агрегированных типов данных языка EXPRESS

B.5 Пример отображения EXPRESS на UML для подтипов

Следующий фрагмент на языке EXPRESS, содержащий объявление SUBTYPE и перечисляемый тип данных, может дать результат в виде UML-диаграммы, показанной на рисунке B.5, и XMI-файла ОМД, аннотированного на рисунке B.6.

```
TYPE fish_type = ENUMERATION OF ( ... );
END_TYPE;

ENTITY pond;
END_ENTITY;

ENTITY fish_pond;
SUBTYPE OF (pond);
fish : SET[1:?] OF fish_type;
END_ENTITY;
```

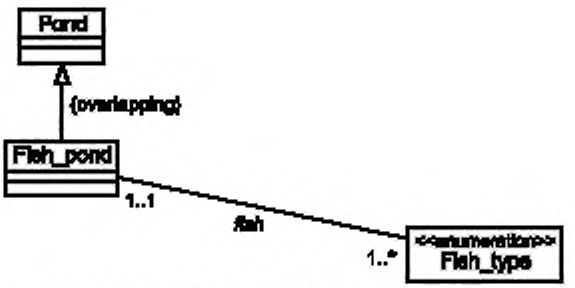

Рисунок B.5 — Пример UML-диаграммы, показывающей классы UML, отображенные из объявления SUBTYPE на языке EXPRESS

```
<Foundation.Core.Class xmi.id= "xmi.2" <
                       xmi.uuid="-63-14-84-52-2d157f:e7e0cb8efa:-7fff">
  <Foundation.Core. ModelElement.name>Pond</Foundation.Core.ModelElement.name>
  ...
  <Foundation.Core.GeneralizableElement.specialization>
    <Foundation.Core.Generalization xmi.idref="xmi.3"/>
  </Foundation.Core.GeneralizableElement.specialization>
</Foundation.Core.Class>

<Foundation.Core.Class xmi.id= "xmi.4"
                       xmi.uuid="-63-14-84-52-2d157f:e7e0cb8efa:-7ffd">
  <Foundation.Core. ModelElement.name>Fish_pond</Foundation.Core.ModelElement.name>
  ...
  <Foundation.Core.GeneralizableElement.generalization>
    <Foundation.Core.Generalization xmi.idref="xmi.3"/>
  </Foundation.Core.GeneralizableElement.generalization>
</Foundation.Core.Class>

<Foundation.Core.Generalization xmi.id="xmi.3"
                                xmi.uuid="-63-14-84-52-2d157f:e7e0cb8efa:-7ffb">
  ...
  <Foundation.Core.Generalization.child>
    <Foundation.Core.GeneralizableElement xmi.idref="xmi.4"/>
  </Foundation.Core.Generalization.child>
  <Foundation.Core.Generalization.parent>
    <Foundation.Core.GeneralizableElement xmi.idref="xmi.2"/>
  </Foundation.Core.Generalization.parent>
</Foundation.Core.Generalization>
```

Рисунок B.6 — Пример файла XMI 1.0 в формате UML 1.3, полученного из объявления SUBTYPE на языке EXPRESS

B.6 Пример отображения EXPRESS на UML для выбираемого типа данных

Следующий фрагмент на языке EXPRESS, содержащий выбираемый тип данных, может дать результат в виде UML-диаграммы, показанной на рисунке B.7.

```
ENTITY person;
END_ENTITY;

ENTITY organization;
END_ENTITY;

TYPE person_or_organization = SELECT( person, organization );
END_TYPE;
```

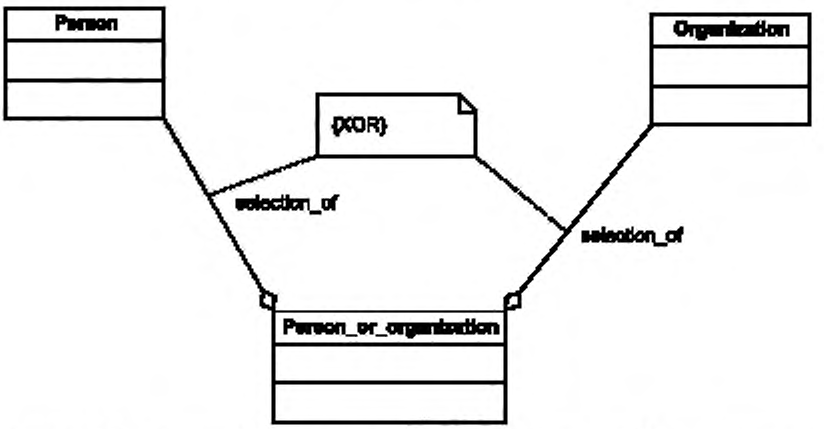

Рисунок B.7 — Пример UML-диаграммы, показывающей классы UML, отображенные из выбираемого типа данных языка EXPRESS

B.7 Пример отображения EXPRESS на UML для определенного типа данных

Следующий фрагмент на языке EXPRESS, содержащий два определенных типа данных, может дать результат в виде UML-диаграммы, показанной на рисунке B.8.

```
TYPE any_int = INTEGER;
END_TYPE;

TYPE positive_int = any_int;
WHERE
greater_than_zero : SELF > 0;
END_TYPE;
```

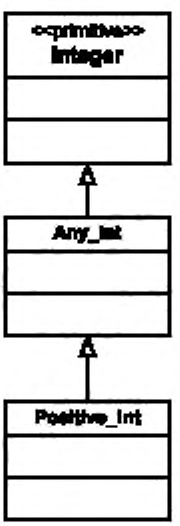

Рисунок B.8 — Пример UML-диаграммы, показывающей классы UML, полученные из определенного типа данных языка EXPRESS

B.8 Пример отображения EXPRESS на UML для вычисляемого атрибута

Следующий фрагмент на языке EXPRESS, содержащий вычисляемый атрибут, может дать результат в виде UML-диаграммы, показанной на рисунке B.9.

```
ENTITY Something;
DERIVE
b : Another := ... ;
END_ENTITY;

ENTITY Another;
END_ENTITY;
```

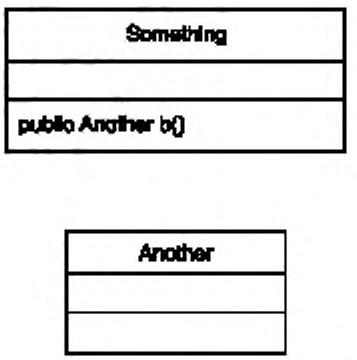

Рисунок B.9 — Пример UML-диаграммы, показывающей UML-операцию, полученную из вычисляемого атрибута языка EXPRESS

Приложение С
(справочное)

Представление на языке EXPRESS частей метамодели обмена UML

В данном приложении представлена EXPRESS-схема, представляющая релевантные части метамодели обмена UML, определенной в [4]. Конструкции UML представлены на языке EXPRESS следующим образом:

- типы данных UML Boolean, Integer и String представлены типами данных BOOLEAN, INTEGER и STRING языка EXPRESS;
- каждое UML-перечисление представлено перечисляемым типом данных языка EXPRESS;
- каждый класс UML представлен объектным типом данных языка EXPRESS;
- каждое UML-обобщение представлено как SUBTYPE языка EXPRESS;
- каждый атрибут UML представлен явным атрибутом языка EXPRESS;
- каждая UML-ассоциация представлена явным атрибутом языка EXPRESS с инверсным атрибутом, представляющим второй конец UML-ассоциации.

```
(* Представление на языке EXPRESS частей метамодели обмена UML, *)
(* определенной в UML 1.4 *)
(*
  К некоторым идентификаторам языка EXPRESS добавлены префиксы в виде строки UML. Наименования этих
  идентификаторов из метамодели обмена UML потребовалось изменить из-за ограничений в языке EXPRESS
  (например, если они совпадали с зарезервированными словами языка EXPRESS).
*)
SCHEMA iso_10303_25_representation_of_uml_interchange_metamodel;

  TYPE AggregationKind = ENUMERATION OF
    (none,
     UML_aggregate,
     composite);
  END_TYPE; - - AggregationKind

  TYPE CallConcurrencyKind = ENUMERATION OF
    (sequential,
     guarded,
     concurrent);
  END_TYPE; - - CallConcurrencyKind

  TYPE ChangeableKind = ENUMERATION OF
    (changeable,
     frozen,
     addonly);
  END_TYPE; - - ChangeableKind

  TYPE Name = STRING;
  END_TYPE; - - Name

  TYPE OrderingKind = ENUMERATION OF
    (unordered,
     ordered,
     sorted);
  END_TYPE; - - OrderingKind

  TYPE ParameterDirectionKind = ENUMERATION OF
    (out);
  END_TYPE; — ParameterDirectionKind

  TYPE ScopeKind = ENUMERATION OF
    (instance,
     UML_classifier);
  END_TYPE; - - ScopeKind
```

```
TYPE VisibilityKind = ENUMERATION OF
   (public,
    protected,
    private);
END_TYPE; - - VisibilityKind
ENTITY Association
   SUBTYPE OF (Relationship, GeneralizableElement);
      connection : LIST [2:?] OF AssociationEnd;
END_ENTITY; - - Association

ENTITY AssociationEnd
   SUBTYPE OF (ModelElement);
      isNavigable    : BOOLEAN;
      ordering       : OrderingKind;
      aggregation    : AggregationKind;
      targetScope    : ScopeKind;
      multiplicity   : Multiplicity;
      changeability  : ChangeableKind;
      UML_type       : Classifier;
   INVERSE
      of_association : Association FOR connection;
END_ENTITY; - - AssociationEnd

ENTITY Attribute
   SUBTYPE OF (StructuralFeature);
END_ENTITY; - - Attribute

ENTITY BehavioralFeature
   ABSTRACT SUPERTYPE OF (ONEOF (Operation))
   SUBTYPE OF (Feature);
      isQuery : BOOLEAN;
   INVERSE
      has     : SET[0:?] OF Parameter FOR behavioralFeature;
END_ENTITY; - - BehavioralFeature

ENTITY Class
   SUBTYPE OF (Classifier);
      isActive : BOOLEAN;
END_ENTITY; - - Class

ENTITY Classifier
   ABSTRACT SUPERTYPE OF (ONEOF (Class, Datatype))
   SUBTYPE OF (GeneralizableElement, Namespace);
INVERSE
   typeof_AssociationEnd     : SET[0:?] OF AssociationEnd FOR UML_type;
   typeof_Parameter          : SET[0:?] OF Parameter FOR UML_type;
   typeof_StructuralFeature  : SET[0:?] OF StructuralFeature FOR UML_type;
   owns                      : SET[0:?] OF Feature FOR owner;
END_ENTITY; - - Classifier

ENTITY Datatype
   SUBTYPE OF (Classifier);
END_ENTITY; - - Datatype

ENTITY Element
   ABSTRACT SUPERTYPE OF (ONEOF (ModelElement));
END_ENTITY; - - Element

ENTITY ElementImport;
      visibility      : VisibilityKind;
      UML_alias       : Name;
      importedElement : ModelElement;
      package         : Package;
END_ENTITY; - - ElementImport
```

```
ENTITY Feature
   ABSTRACT SUPERTYPE OF (ONEOF (StructuralFeature, BehavioralFeature))
   SUBTYPE OF (ModelElement);
      ownerScope : ScopeKind;
      owner      : Classifier;
END_ENTITY; - - Feature

ENTITY GeneralizableElement
   ABSTRACT SUPERTYPE OF (ONEOF (Classifier, Association, Package))
   SUBTYPE OF (ModelElement);
      isRoot     : BOOLEAN;
      isLeaf     : BOOLEAN;
      isAbstract : BOOLEAN;
   INVERSE
      parents    : SET[0:?] OF Generalization FOR child;
      children   : SET[0:?] OF Generalization FOR parent;
END_ENTITY; - - GeneralizableElement

ENTITY Generalization
   SUBTYPE OF (Relationship);
      discriminator : Name;
      parent        : GeneralizableElement;
      child         : GeneralizableElement;
END_ENTITY; - - Generalization

ENTITY ModelElement
   ABSTRACT SUPERTYPE OF (ONEOF (Feature, Namespace, GeneralizableElement,
      Parameter, AssociationEnd, Relationship))
   SUBTYPE OF (Element);
      name          : Name;
      visibility    : VisibilityKind;
      isSpecification : BOOLEAN;
      namespace     : Namespace;
   INVERSE
      imported_by   : SET[0:?] OF ElementImport FOR importedElement;
END_ENTITY; - - ModelElement

ENTITY Multiplicity;
      range : Multiplicityrange;
   INVERSE
      size_of : SET[0:?] OF StructuralFeature FOR multiplicity;
END_ENTITY; - - Multiplicity

ENTITY Multiplicityrange;
      lower : INTEGER;
      upper : INTEGER;
   INVERSE
      range_of : SET[1:?] OF Multiplicity FOR range;
END_ENTITY; - - Multiplicityrange

ENTITY Namespace
   ABSTRACT SUPERTYPE OF (ONEOF (Classifier, Package))
   SUBTYPE OF (ModelElement);
   INVERSE
      elements : SET[0:?] OF ModelElement FOR namespace;
END_ENTITY; - - Namespace

ENTITY Operation
   SUBTYPE OF (BehavioralFeature);
      concurrency : CallConcurrencyKind;
      isRoot      : BOOLEAN;
      isLeaf      : BOOLEAN;
      isAbstract  : BOOLEAN;
      specification : STRING;
END_ENTITY; - - Operation
```

```
    ENTITY Package
      SUPERTYPE OF (ONEOF (UML_Model))
      SUBTYPE OF (Namespace, GeneralizableElement);
    INVERSE
      imports : BAG[0:?] OF ElementImport FOR package;
    END_ENTITY; - - Package

    ENTITY Parameter
      SUBTYPE OF (ModelElement);
          UML_type          : Classifier;
          kind              : ParameterDirectionKind;
          behavioralFeature : BehavioralFeature;
    END_ENTITY; - - Parameter

    ENTITY Relationship
      ABSTRACT SUPERTYPE OF (ONEOF (Generalization, Association))
      SUBTYPE OF (ModelElement);
    END_ENTITY; - - Relationship
    ENTITY StructuralFeature
      ABSTRACT SUPERTYPE OF (ONEOF (Attribute))
      SUBTYPE OF (Feature);
          multiplicity    : Multiplicity;
          changeability   : ChangeableKind;
          targetScope     : ScopeKind;
          ordering        : OrderingKind;
          UML_type        : Classifier;
    END_ENTITY; - - StructuralFeature

    ENTITY UML_Model
      SUBTYPE OF (Package);
    END_ENTITY; - - UML_Model

END_SCHEMA; - - iso_10303_25_representation_of_uml_interchange_metamodel
```

Приложение D
(справочное)

Выдержки из Спецификации UML 1.4

В данном приложении приведены диаграммы и текстовые выдержки из Спецификации UML 1.4 [4], поясняющие отображение языка EXPRESS на метамодель обмена UML. Эти выдержки воспроизведены с разрешения OMG, обладающей авторским правом.

D.1 Метамодель обмена UML

Пользователи, незнакомые с метамоделью обмена UML, должны ознакомиться с обзором, приведенным в [4, раздел 5.1].

На рисунке D.1 представлена диаграмма наивысшего уровня метамодели обмена UML. Большинство аспектов на данной диаграмме используется при отображении языка EXPRESS на метамодель обмена UML. Объектный тип данных языка EXPRESS отображается на класс UML, EXPRESS-схема отображается на подкласс UML Namespace, некоторые явные атрибуты языка EXPRESS отображаются на атрибут UML, а подтип данных языка EXPRESS отображается на UML-обобщение.

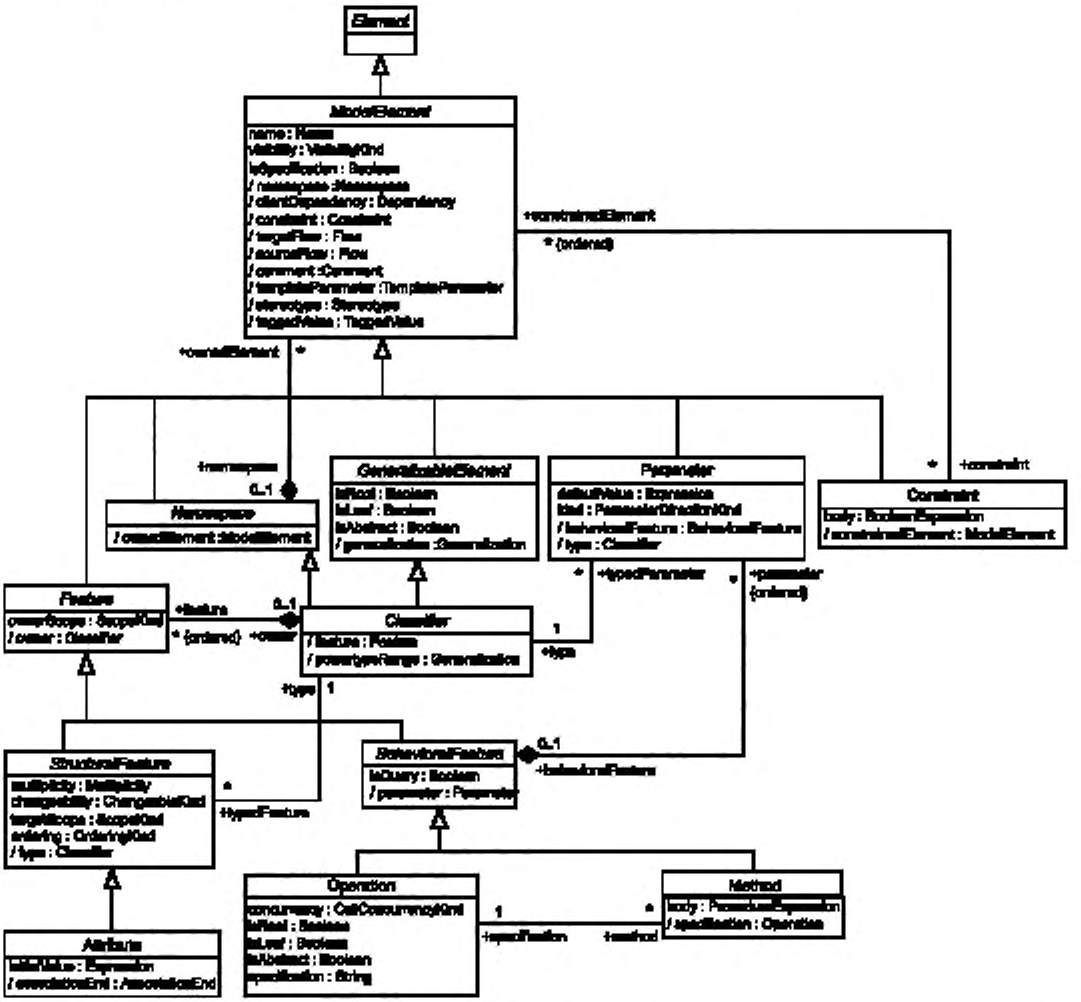

Рисунок D.1 — Главный пакет — Основа (рисунок 5-2 из [4])

На рисунке D.2 представлена диаграмма метамодели обмена UML, отражающая взаимосвязи. Большинство аспектов на этой диаграмме используется при отображении языка EXPRESS на метамодель обмена UML, за исключением объектов UML Flow и UML AssociationClass. В языке EXPRESS нет понятий, эквивалентных UML-ассоциации или объекту UML AssociationEnd. Однако, некоторые атрибуты языка EXPRESS, областью определения которых являются именованные типы данных языка EXPRESS, отображаются на UML-ассоциации.

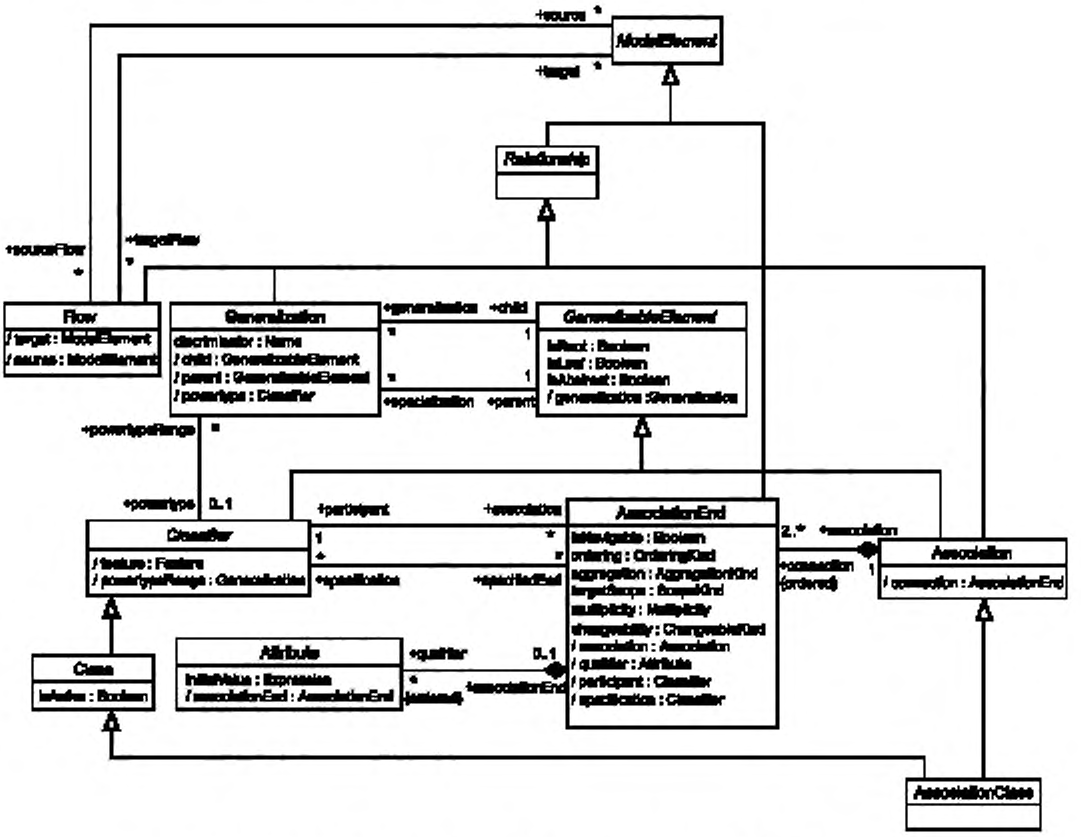

Рисунок D.2 — Главный пакет — Взаимосвязи (рисунок 5-3 из [4])

На рисунке D.3 представлена диаграмма метамодели обмена UML, отражающая классификаторы. При отображении языка EXPRESS на метамодель обмена UML используются только класс UML, тип данных UML и UML-перечисление.

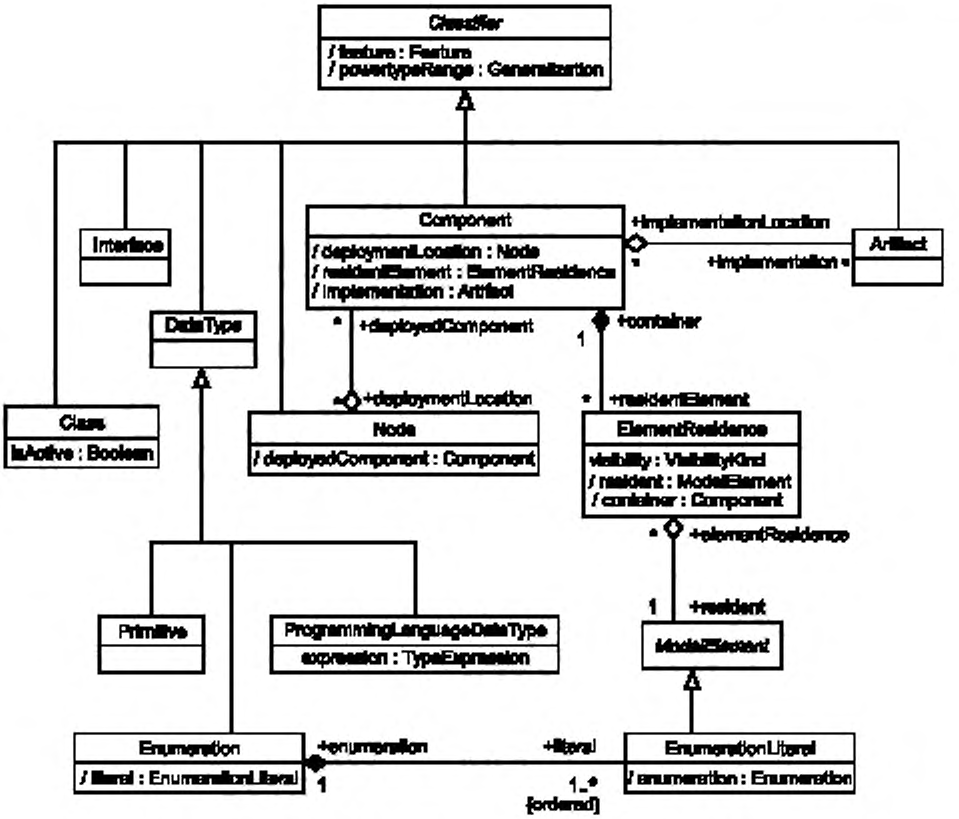

Рисунок D.3 — Главный пакет — Классификаторы (рисунок 5-5 из [4])

На рисунке D.4 представлена диаграмма метамодели обмена UML, отражающая управление моделью. Большинство аспектов данной диаграммы используется при отображении языка EXPRESS на метамодель обмена UML, за исключением объекта UML Subsystem. Управление моделью UML играет ту же роль, что и EXPRESS-схемы, связанные через спецификацию интерфейсов языка EXPRESS.

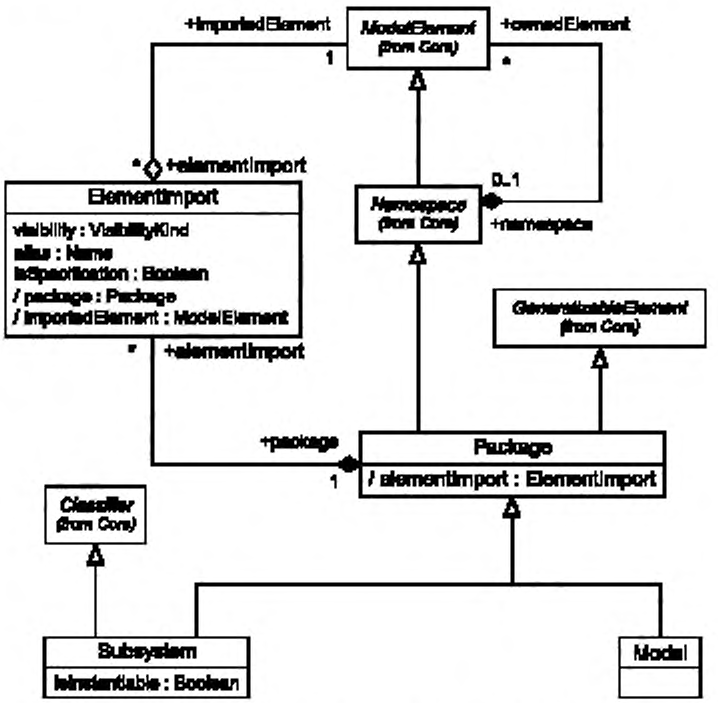

Рисунок D.4 — Управление моделью (рисунок 5-21 из [4])

D.2 UML-ассоциация

Приведенный ниже текст взят непосредственно из [4, пункт 2.5.2].

Ассоциация

Ассоциация определяет семантическую взаимосвязь между классификаторами. Экземплярами ассоциации является множество кортежей, связанных с экземплярами классификаторов. Значение каждого кортежа может появиться не более одного раза.

Ассоциация в метамодели является объявлением семантической взаимосвязи между классификаторами, например между классами. У ассоциации есть не менее двух концов связи (объектов AssociationEnd). Каждый конец соединен с классификатором. Классификатор может быть соединен с несколькими объектами AssociationEnd в одной ассоциации. Ассоциация представляет множество соединений между экземплярами классификаторов. Экземпляром ассоциации является связь (объект Link), которая представляет собой кортеж экземпляров, взятых из соответствующих классификаторов.

Атрибут

name (наименование): Наименование ассоциации, которое в совокупности со связанными с ним классификаторами должно быть уникальным в рамках включающего его пространства имен (обычно в рамках пакета).

Ассоциация

connection (соединение): Ассоциация содержит по крайней мере два объекта AssociationEnd, каждый из которых представляет соединение ассоциации с классификатором. Каждый объект AssociationEnd определяет ряд свойств, которые должны быть выполнены для того, чтобы данная взаимосвязь была правомерной. Основная часть структуры ассоциации определяется ее объектами AssociationEnd. Классификаторы, принадлежащие ассоциации, связаны с объектами AssociationEnd посредством ассоциации ролевого имени участника.

Стереотип

implicit (неявный): Стереотип «implicit» применяется к ассоциации, указывая на то, что ассоциация не является явной, а только концептуальной.

Стандартное ограничение

xor (исключающее или): Ограничение {xor} применяется к совокупности ассоциаций, указывая на то, что в рамках данной совокупности только одна ассоциация является явной для каждого ассоциированного экземпляра. Xor является ограничением «исключающее или» (но не «включающее или».

Отмеченное значение

persistence (персистентность): Персистентность обозначает постоянство состояния ассоциации, отмечая его как кратковременное (состояние ассоциации аннулируется при уничтожении данного экземпляра) или неизменное (состояние ассоциации не аннулируется при уничтожении данного экземпляра).

Наследуемое свойство

Ассоциация является объектом GeneralizableElement. Ассоциацией-потомком наследуется следующий элемент:

connection (соединение): Потомок должен иметь такое же число концов, что и предок. Каждый участвующий класс должен быть потомком участвующего класса на той же позиции в предке. Если Ассоциация является объектом AssociationClass, то ее свойства класса (атрибуты, операции и т. д.) наследуются. Разнообразные другие свойства могут изменяться у потомка. Данная спецификация, возможно, будет детализирована в UML 2.0.

Ненаследуемые свойства

isRoot, isLeaf, isAbstract: Они не являются наследуемыми по своей сущности, поскольку определяют структуру обобщения.

name: Каждый элемент модели имеет уникальное наименование.

Приложение Е
(справочное)

Краткое изложение отображения

В таблице Е.1 содержится краткое изложение отображения из языка EXPRESS на UML с использованием фрагментов, взятых из EXPRESS-схем примера в В.2, и полученного результата на UML. Файл из примера, соответствующий XMI 1.2, содержит представления большинства конструкций, имеющихся в схемах примера.

Примечание — Следующие конструкции не включены в таблицу Е1, так как инструментарий UML, используемый для создания XMI-файла, не поддерживает данные конструкции:

- ONE OF;
- REFERENCE FROM;
- USE FROM одиночный объект и AS для его переименования;
- литералы перечисляемого типа.

Таблица Е.1 — Краткое изложение отображения EXPRESS на UML на примерах

Фрагмент на языке EXPRESS	Результирующий фрагмент метамодели обмена UML 1.4
SCHEMA context_schema;	UML-пакет, в котором атрибут UML ModelElement.name имеет значение 'Context_schema', определенное в рамках UML-модели, представляющей обзор данных системы
SCHEMA context_schema; USE FROM s2_schema; SCHEMA s2_schema;	1. UML-пакет с именем 'S2_schema', содержащий UML-представления всех элементов языка EXPRESS из 's2_schema'. 2. Объект UML ElementImport, импортирующий UML-пакет 'S2_schema' в UML-пакет 'Context_schema'
SCHEMA context_schema; REFERENCE FROM s2_schema; SCHEMA s2_schema;	REFERENCE FROM дает такой же результат на UML, что и USE FROM
SCHEMA context_schema; USE FROM s3_schema (entity1 AS renentity1); SCHEMA s3_schema;	1. UML-пакет с именем 'S3_schema', содержащий UML-представления элементов языка EXPRESS из 's3_schema', видимых в 'context_schema'. UML-пакет определен в рамках UML-модели, представляющей обзор данных системы. 2. Объект UML ElementImport 'Entity1' и что-либо видимое в 'context_schema' в UML-пакете с именем 'Context_schema'. 3. Атрибут UML ElementImport.alias переименовывает 'Entity1' в 'Renentity1'
SCHEMA context_schema; ENTITY entity2; ENTITY entity1 SUBTYPE OF (entity2);	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity2', пространством имен класса является UML-пакет с именем 'Context_ schema'. 2. Класс UML с атрибутом Class.name, имеющим значение 'Entity1', пространством имен класса является UML-пакет с именем 'Context_ schema'. 3. UML-обобщение с классом UML 'Entity1' в качестве потомка и классом UML 'Entity2' в качестве предка. 4. Ограничение перекрытия UML на ' Entity1' и любой другой класс UML, представляющий объектный тип данных языка EXPRESS и являющийся подтипом 'entity2'
SCHEMA context_schema; ENTITY entity2 ABSTRACT SUPERTYPE;	Класс UML с атрибутом UML Class.name, имеющим значение 'Entity2', пространством имен класса UML является UML-пакет с именем 'Context_schema'. Атрибут isAbstract класса UML имеет значение 'true' (истина)
In ENTITY entity1, SUPERTYPE OF (ONEOF (entity4, entity5)	Если 'entity4' и 'entity5' являются только подтипами 'entity1', то ограничение непересечения UML распространяется на все классы UML, представляющие объектный тип данных языка EXPRESS, который является подтипом 'entity1'

Продолжение таблицы Е.1

Фрагмент на языке EXPRESS	Результирующий фрагмент метамодели обмена UML 1.4
TYPE type1 = SELECT (entity1, type2);	1. Класс UML с атрибутом Class.name, имеющим значение 'Type1'. 2. Класс UML с атрибутом Class.name, имеющим значение 'Entity1' . 3. Класс UML с атрибутом Class.name, имеющим значение 'Type2'. 4. Агрегированная UML-ассоциация между классами UML 'Entity1', 'Type2' с объектом AssociationEnd с именем 'none' и 'Type1' с объектом AssociationEnd с именем 'aggregate'. 5. Ограничение ИЛИ UML между UML-ассоциациями агрегаций, соединяющими классы UML 'Entity1' и 'Type2' с классом UML 'Type1'
TYPE type2 = ENUMERATION OF (red, white);	UML-перечисление с атрибутом UML Enumeration.name, имеющим значение 'Type2', и объектами UML EnumerationLiteral с именами 'red' и 'white'
TYPE type3 = STRING	Если базовый тип определенного типа данных не является выбираемым или перечисляемым типом данных: 1. Класс UML с атрибутом Class.name, имеющим значение 'Type3'. 2. UML-обобщение с классом UML 'Type3' в качестве потомка и UML-представление базового типа данных в качестве предка
ENTITY entity10; a1 : INTEGER;	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity10'. 2. Атрибут UML с параметрами UML Attribute.name, имеющим значение 'a1', и UML Attribute.type, имеющим значение UML Integer, а пространством имен атрибута UML 'a1' является класс UML 'Entity10'
ENTITY entity5; ENTITY entity4; a2 : entity5;	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity5'. 2. Класс UML с атрибутом Class.name, имеющим значение 'Entity4'. 3. UML-ассоциация с атрибутом Association.name, имеющим значение 'a2', объектами AssociationEnd являются классы UML 'Entity5' и 'Entity4', значение кратности задано как 1..1 для 'Entity5' и 0..* для 'Entity4', значением атрибута AssociationEnd.name является 'a2' для 'Entity5'
ENTITY entity5; INVERSE ia1 : SET[1:?] OF entity4 FOR a2; ENTITY entity4; a2 : entity5;	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity5'. 2. Класс UML с атрибутом Class.name, имеющим значение 'Entity4'. 3. UML-ассоциация с атрибутом Association.name, имеющим значение 'a2', объектами AssociationEnd являются классы UML 'Entity5' и 'Entity4', значение кратности задано как 1..1 для 'Entity4' и 1..* для 'Entity5', значением атрибута AssociationEnd.name является 'ia1' для 'Entity4' и 'a2' для 'Entity5'
ENTITY entity2; a5 : LIST OF type1;	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity2'. 2. Многозначный атрибут UML с параметрами UML Attribute.name, имеющим значение 'a5', и UML Attribute.type, имеющим значение класс UML 'Type1', пространством имен атрибута UML 'a1' является класс UML 'Entity2', атрибут UML имеет кратность 0..* и признак 'ordered'
ENTITY entity4; ENTITY entity2; a6 : LIST OF UNIQUE entity4;	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity2' 2. UML-ассоциация с атрибутом Association.name, имеющим значение 'a6', объектами AssociationEnd являются классы UML 'Entity4' и 'Entity2', значение кратности задано как 0..* для 'Entity2' и 0..* для 'Entity4', класс UML 'Entity4' имеет признак 'ordered', значением атрибута AssociationEnd.name является 'a6' для 'Entity4'
ENTITY entity10; ENTITY entity1; a11 : SET OF entity10;	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity1'. 2. UML-ассоциация с атрибутом Association.name, имеющим значение 'a11', объектами AssociationEnd являются классы UML 'Entity1' и "Entity10', значение кратности задано как 0..* для 'Entity10' и 0..* для 'Entity1', значением атрибута AssociationEnd.name является 'a11' для 'Entity10'
ENTITY entity1; a7 : SET OF STRING;	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity1'. 2. Класс UML с атрибутом Class.name, имеющим значение 'Set-of-string'. 3. UML-ассоциация между классами UML 'Entity1' и 'Set-of-string' с атрибутом UML Association.name, имеющим значение 'a7', значение кратности задано как 0..* для конца 'Entity1' и 1..1 для конца 'Set-of-string'

Окончание таблицы Е.1

Фрагмент на языке EXPRESS	Результирующий фрагмент метамодели обмена UML 1.4
ENTITY entity1; a7 : SET OF STRING;	4. Агрегированная UML-ассоциация между классом UML 'Set-of-string' и типом данных UML String
ENTITY entity4; a8 : LIST OF STRING;	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity4'. 2. Класс UML с атрибутом Class.name, имеющим значение 'List-of-string'. 3. UML-ассоциация между классами UML 'Entity1' и 'List-of-string' с атрибутом UML Association.name, имеющим значение 'a8', значение кратности задано как 0..* для конца 'Entity1' и 1..1 для конца 'List-of-string'. 4. Многозначный атрибут UML в классе UML 'List-of-string', типом которого является тип данных UML String, заданы значение кратности 0..* и признак 'ordered'
TYPE type4 = LIST OF STRING;	1. Класс UML с атрибутом Class.name, имеющим значение 'Type4'. 2. Класс UML с атрибутом Class.name, имеющим значение 'List-of-string'. 3. Многозначный атрибут UML в классе UML 'List-of-string', типом которого является тип данных UML String, заданы значение кратности 0..* и признак 'ordered'. 4. UML-обобщение между классами UML 'Type4' и 'List-of-string', где 'Type4' является потомком, а 'List-of-string' — предком
ENTITY entity2; DERIVE da1 : REAL := ...	1. Класс UML с атрибутом Class.name, имеющим значение 'Entity2. 2. UML-операция с атрибутом UML Operation.name, имеющим значение 'da1', пространством имен является класс UML 'Entity2', а значением параметра isQuery является 'true'. 3. Параметр UML с атрибутами Parameter.parameter, имеющим значение 'da1', и Parameter.type, имеющим значение типа данных UML 'Double'

Отображения для каждой из основных конструкций языка EXPRESS определены в следующих разделах настоящего стандарта:

- отображения объявления схемы — в 5.4;
- отображения спецификации интерфейса — в 5.4;
- отображения объектного типа данных — в 5.5;
- отображения объявления подтипов — в 5.6;
- отображения инверсного атрибута — в 5.7;
- отображения вычисляемого атрибута — в 5.8;
- отображения перечисляемых типов данных — в 5.9.1;
- отображения выбираемых типов данных — в 5.9.2;
- отображения определенных типов данных, не являющихся выбираемыми или перечисляемыми типами данных — в 5.9.3.

Отображения для каждой из конструкций области определения явных атрибутов языка EXPRESS определены в следующих разделах настоящего стандарта:

- отображения области определения простых типов данных — в 5.7.1 и 5.3;
- отображения области определения одиночных именованных типов данных — в 5.7.2 и 5.5 или 5.9;
- отображения области определения типа данных SET, включающего именованные типы данных, — в 5.7.2 и 5.5 или 5.9;
- отображения области определения типа данных LIST OF UNIQUE, включающего именованные типы данных, — в 5.7.2 и 5.5 или 5.9;
- отображения области определения типа данных BAG, включающего именованные типы данных, — в 5.7.3 и 5.5 или 5.9;
- отображения области определения неуникального типа данных LIST, включающего именованные типы данных, — в 5.7.3 и 5.5 или 5.9;
- отображения области определения типа данных ARRAY, включающего именованные типы данных, — в 5.7.4 и 5.5 или 5.9;
- отображения области определения агрегированных типов данных, включающих простые типы данных, — в 5.7.4, 5.10.2 и 5.10.3;
- отображения области определения агрегированных типов данных, включающих агрегированные типы данных, — в 5.7.4, 5.10.1 и 5.10.3.

Приложение F
(справочное)

Техническое обсуждение

В данном приложении обсуждается взаимосвязь языка EXPRESS с другими стандартами и работали, проводимыми OMG и связанными с XML.

F.1 EXPRESS в MOF

Метаобъектные средства OMG описаны в разделе 2 спецификации MOF, версия 1.4 следующим образом:

UML и MOF базируются на концептуальной многоуровневой архитектуре метамодели, в которой элементы на данном концептуальном уровне описывают элементы на следующем нижележащем уровне. Например,

- мета-метамодель MOF — это язык, используемый для определения метамодели UML;
- метамодель UML — это язык, используемый для определения моделей UML, и
- модель UML — это язык, который определяет аспекты компьютерной системы.

Таким образом, метамодель UML может быть описана как «экземпляр» мета-метамодели MOF, а модель UML — как «экземпляр» метамодели UML. Однако эти логические объекты не обязательно должны существовать в одной и той же области определения представлений типов данных и их значений. Данный подход иногда называют несвязанным метамоделированием.

Основным аспектом подхода MOF к управлению метаданными является расширяемость. Цель заключается в создании структуры, поддерживающей любой вид метаданных и позволяющей добавлять при необходимости новые виды метаданных. Для достижения данной цели MOF имеет многоуровневую архитектуру метаданных, базирующуюся на классической четырехуровневой архитектуре метамоделирования, популярной среди сообществ, занимающихся разработкой стандартов, таких как ИСО и CDIF. Основной особенностью обеих архитектур метаданных, как классической, так и MOF, является уровень мета-метамоделирования, который связывает вместе метамодели и модели.

Архитектура метаданных MOF иллюстрируется примером на рисунке 2-2.

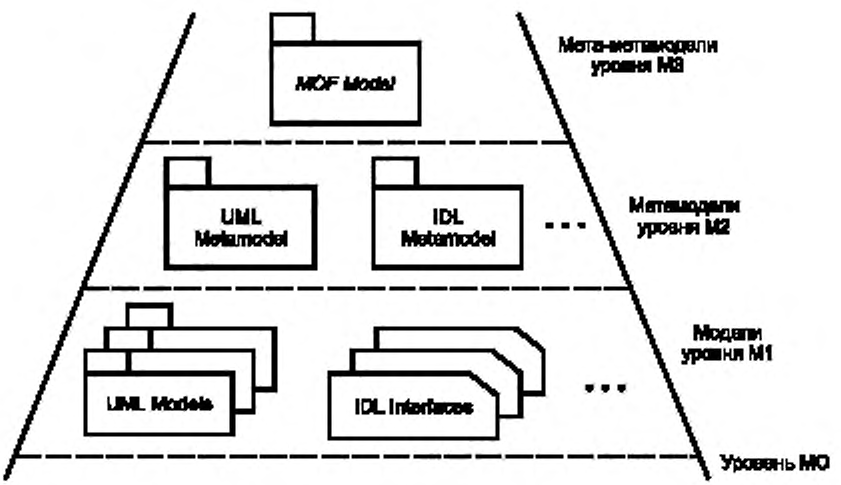

Рисунок F.1 — Рисунок 2-2 «Архитектура метаданных MOF» из Спецификации MOF 1.4

XMI базируется на MOF, и документ в формате XMI может содержать все четыре уровня данной архитектуры. В качестве следующего шага в развитии и гармонизации взаимосвязи между стандартами EXPRESS и OMG определение EXPRESS на уровне М2 архитектуры, использующей MOF на уровне М3, должно привести к возможности использования MOF-совместимых репозиториев для управления языком EXPRESS и EXPRESS-схемами напрямую, без их отображения на UML. При этом EXPRESS-схемы могли бы управляться на уровне М1, а XMI имел бы возможность обмениваться этими тремя уровнями между MOF/XMI-совместимыми репозиториями. Следует отметить, что данная деятельность стала бы первой попыткой стандартизации метамодели языка EXPRESS. В ИСО 10303-22 определен словарь данных для EXPRESS-схем в среде интерфейсов доступа к данным, который мог бы использоваться в качестве отправной точки при разработке метамодели EXPRESS с использованием MOF.

Если EXPRESS-схемами можно будет управлять на уровне M1 новой архитектуры, то экземплярами объектов языка EXPRESS можно будет управлять на уровне M0. Следовательно, результатом стандартизованного отображения EXPRESS на MOF будет способность обмениваться экземплярами объектов языка EXPRESS, EXPRESS-схемами, метамоделью MOF языка EXPRESS и самими MOF в одном XMI-документе. Поскольку не требуется, чтобы все уровни присутствовали в каком-либо конкретном XMI-документе, то возможно обеспечить обмен XMI-совместимыми схемами и данными с использованием XML способом, аналогичным описанному в ИСО 10303-28. Форма могла бы быть в основном похожа на обмен схемами и динамическое связывание в ИСО 10303-28. Стандартизация представления в MOF языка EXPRESS должна быть проработана в качестве потенциального приоритета для последующей редакции настоящего стандарта.

F.2 Обработка ограничений и выражений

Как указано в 5.1, некоторые конструкции языка EXPRESS находятся вне области применения установленного отображения. Двумя важными конструкциями, не включенными в установленные отображения, являются: 1) ограничения супертипов и 2) правила и выражения. OMG определила язык описания ограничений для UML, названный языком описания объектных ограничений (OCL). В будущих редакциях настоящего стандарта может появиться возможность включить отображение ограничения языка EXPRESS на OCL. Определение такого перевода представляется большой исследовательской задачей. Однако на сегодняшний день не проведено никакого детального исследования совместимости между этими языками, и поэтому подобный перевод еще не готов для стандартизации.

Приложение G
(справочное)

Сценарии применения

В данном приложении рассмотрены сценарии применения, в которых могут быть использованы реализации настоящего стандарта.

G.1 Реализация прикладного протокола

Одним из возможных сценариев применения настоящего стандарта является реализация прикладного протокола (ПП) с использованием UML-ориентированных автоматизированных средств разработки программного обеспечения. В данный сценарий включены следующие этапы:

1. Завершение процесса разработки ПП, приводящего в результате к EXPRESS-схеме, которая должна использоваться в соответствующих реализациях.

2. Использование программного приложения, соответствующего настоящему стандарту, для преобразования EXPRESS-схемы в XML-документ, основанный на метамодели обмена UML и версии XMI, поддерживаемой конкретным UML-ориентированным средством моделирования программного обеспечения.

3. Интерпретирование XMI-файла в UML-ориентированном средстве моделирования программного обеспечения.

4. Генерация UML-модели, описывающей всю систему программного обеспечения, с использованием UML-ориентированного средства моделирования программного обеспечения.

5. Использование UML-ориентированного средства моделирования программного обеспечения для автоматизированной генерации кода на желаемом языке программирования (например, Java).

6. Завершение реализации ПП на выбранном языке программирования.

В случае, если реализацией ПП является прикладная программа для обмена данными, то UML-модель, сгенерированная UML-ориентированным средством, описывает внутренние действия программной системы. Может быть так, что данное использование UML при разработке ПП будет невидимым для пользователей данной реализации, которые видят только файлы в формате ИСО 10303-21, которые воспринимаются и вырабатываются данной реализацией. Одним из преимуществ использования автоматизированных средств, подобных описанным в данном сценарии, является улучшение качества системы при ее реализации. Поскольку UML-модель системы программного обеспечения инициируется непосредственно EXPRESS-схемой, то вероятность возникновения ошибок, связанных с правильным отображением конструкций языка EXPRESS будет меньше. На рисунке 1 во введении показан вариант использования такой системы.

G.2 Генерация интегрированной реляционной базы данных, поддерживающей UML и EXPRESS

Вторым возможным сценарием использования настоящего стандарта является реализация, которая основывается на реляционной базе данных, а информационные требования смоделированы разными группами, одна из которых использует язык UML, а другая — EXPRESS. Некоторые UML-ориентированные средства моделирования программного обеспечения могут автоматически генерировать реляционные базы данных и прикладной программный интерфейс, обеспечивающий доступ к базе данных. В этом часть сценария, относящаяся к интеграции двух парадигм моделирования на уровне реляционной базы данных, будет включать следующие этапы:

1. Использование прикладной программы, соответствующей настоящему стандарту, для преобразования EXPRESS-схемы в XML-документ, основанный на метамодели обмена UML и версии XMI, поддерживаемой конкретным UML-ориентированным средством моделирования программного обеспечения.

2. Интерпретирование XMI-файла в UML-ориентированном средстве моделирования программного обеспечения.

3. Интегрирование конструкций UML, полученных из EXPRESS-схемы, с конструкциями, извлеченными из информационной модели, созданной непосредственно в UML-ориентированном средстве моделирования программного обеспечения.

4. Использование UML-ориентированного средства моделирования программного обеспечения для автоматической генерации реляционной базы данных и прикладного программного интерфейса, обеспечивающего доступ к этой базе данных на желаемом языке программирования (например, Java).

На рисунке G.1 показано, как может работать такой интеграционный сценарий.

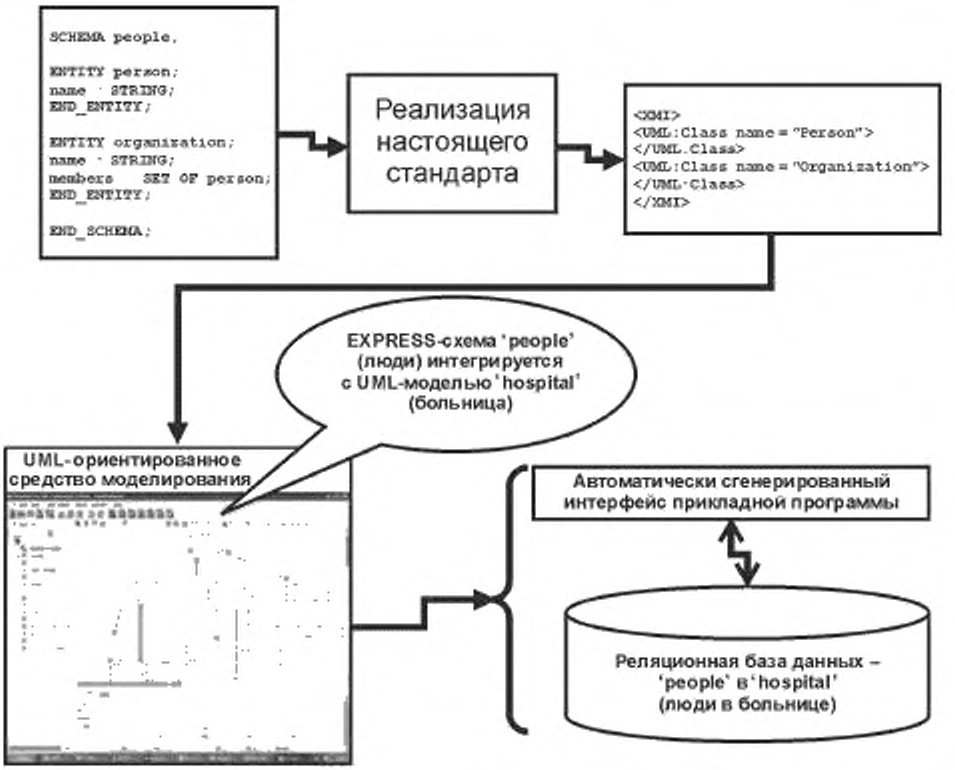

Рисунок G.1 — Интегрированные базы данных EXPRESS и UML

Сценарий интегрирования информационных моделей на языке EXPRESS с UML-моделями не ограничивается генерацией реляционных баз данных.

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов и документов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта, документа	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО/МЭК 8824-1:2002	IDT	ГОСТ Р ИСО/МЭК 8824-1—2001 «Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации»
ИСО 10303-1:1994	IDT	ГОСТ Р ИСО 10303-1—99 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 1. Общие представления и основополагающие принципы»
ИСО 10303-11:1994	IDT	ГОСТ Р ИСО 10303-11—2009 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS»
ИСО 10303-28:2003	—	*
OMG UML 1.5	—	*
OMG XMI 1.2	—	*
MOF 1.4	—	*

* Соответствующий национальный стандарт отсутствует. До его утверждения рекомендуется использовать перевод на русский язык данного международного стандарта, документа. Перевод данного международного стандарта, документа находится в Федеральном информационном фонде технических регламентов и стандартов.

Примечание — В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов:

- IDT — идентичные стандарты.

Библиография

[1] Meta Object Facility Specification Version 1.3. Object Management Group formal/2000-04-03 [cited 2003-02-22]. Available from the World Wide Web: < gost.e-ecolog.ru >

[2] Extensible Markup Language (XML) 1.0 (Second Edition). World Wide Web Consortium Recommendation 6 October 2000 [cited 2003-02-22]. Available from World Wide Web: < gost.e-ecolog.ru >

[3] OMG Unified Modeling Language Specification Version 1.3, Object Management Group Document formal/00-03-01, First Edition, March 2000 [cited 2003-01-28]. Available from World Wide Web: < gost.e-ecolog.ru doc?formal/00-03-01>

[4] OMG Unified Modeling Language Specification Version 1.4, Object Management Group Document formal/01-09-67, September 2001 [cited 2004-02-11]. Available from World Wide Web: < gost.e-ecolog.ru 01-09-67>

[5] OMG XML Metadata Interchange (XMI) Specification Version 1.1, Object Management Group Document formal/ 2000-11-02, November 2000 [cited 2003-01-28]. Available from World Wide Web: < gost.e-ecolog.ru doc?formal/00-11-02>

[6] OMG XML Metadata Interchange (XMI) Specification Version 1.0, Object Management Group Document formal/ 2000-06-01, June 2000 [cited 2003-01-28]. Available from World Wide Web: < gost.e-ecolog.ru doc?formal/00-06-01>

[7] SC4 Supplementary directives — Rules for the structure and drafting of SC4 standards for industrial data. ISO TC184/SC4 N1217, 2001-11-01

УДК 656.072:681.3:006.354 ОКС 25.040.40 П87 ОКСТУ 4002

Ключевые слова: автоматизация производства, средства автоматизации, интеграция систем автоматизации, данные об изделиях, представление данных, обмен данными, методы реализации, язык EXPRESS, язык UML, стандарт XMI

Редактор *Н. В. Авилочкина*
Технический редактор *Н. С. Гришанова*
Корректор *Л. Я. Митрофанова*
Компьютерная верстка *З. И. Мартыновой*

Сдано в набор 23.04.2013. Подписано в печать 07.06.2013. Формат 60×84 $^{1}/_{8}$. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 5,58. Уч.-изд. л. 5,10. Тираж 71 экз. Зак. 658

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru @gostinfo.ru
Набрано и отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256.

gost.e-ecolog.ru