ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 13584-42— 2012

Системы промышленной автоматизации и интеграция

БИБЛИОТЕКА ДЕТАЛЕЙ

Часть 42

Методология описания. Методология структурирования семейств деталей

ISO 13584-42:2010
Industrial automation systems and integration — Parts library — Part 42: Description methodology: Methodology for structuring parts families (IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН АНО «Международная академия менеджмента и качества бизнеса» на основе собственного аутентичного перевода на русский язык стандарта, указанного в пункте 4

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 100 «Стратегический и инновационный менеджмент»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 29 ноября 2012 г. №1704-с

4 Настоящий стандарт идентичен международному стандарту ИСО 13584-42:2010 «Системы промышленной автоматизации и интеграция. Библиотека деталей. Часть 42. Методология описания. Методология структурирования семейств деталей» (ISO 13584-42:2010 «Industrial automation systems and integration — Parts library — Part 42: Description methodology: Methodology for structuring parts families»).

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

5 ВВЕДЕН ВПЕРВЫЕ

Правила применения настоящего стандарта установлены в ГОСТ Р 1.0—2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (gost.ru)

© Стандартинформ, 2014

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 2
3 Термины и определения 2
4 Аббревиатуры 10
5 Описание иерархии характеристических классов продукции 10
5.1 Соотношения между категориями продукции и онтологиями продукции 10
5.2 Соотношения между классами 10
5.3 Одновременное описание характеристических классов продукции и свойств продукции 12
5.4 Применимые и видимые свойства 12
5.5 Назначение иерархии стандартных характеристик 13
5.6 Использование иерархии стандартных характеристик 13
5.7 Свойства со значением, определенным классом 14
5.8 Совместимость ИСО 13584 и МЭК 61360 14
6 Правила создания иерархий характеристических классов продукции 14
6.1 Выбор иерархии характеристических классов 14
6.2 Ассоциация свойств 16
7 Словарные элементы, описывающие свойства продукции 17
7.1 Отображение свойств на общую словарную модель ИСО 13584/МЭК 61360 17
7.2 Атрибуты 17
8 Словарные элементы, описывающие классы продукции 23
8.1 Отображение классов на общую словарную модель ИСО 13584/МЭК 61360 23
8.2 Атрибуты 24
9 Правила управления изменением словаря 31
9.1 Принцип онтологической непрерывности 31
9.2 Пересмотры и версии 32
9.3 Исправление ошибок 34
9.4 Правила управления изменениями 35
9.5 Изменения и атрибуты словаря 40
9.6 Ограничения на эволюцию ссылочных словарей 40
Приложение A (обязательное) Обзор кодов классификации типов неколичественных типов элементов данных (основной класс A) 42
Приложение B (обязательное) Сущность короткого имени 43
Приложение C (обязательное) Компьютерно-интерпретируемые листинги 46
Приложение D (обязательное) Спецификация формата значения 47
Приложение E (обязательное) Регистрация информационного объекта 58
Приложение F (справочное) Подмножество общей словарной схемы МЭК/ИСО, задокументированное в настоящей части ИСО 13584 59
Приложение G (справочное) Обзор основных классов и категорий свойств 156
Приложение H (справочное) Обзор кодов классификации типов количественных типов элементов данных 157
Приложение I (справочное) Диаграмма EXPRESS-G 163
Приложение J (справочное) Частные словари 174
Приложение K (справочное) Информационная поддержка практической реализации 175
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации 175
Библиография 176

Введение

Комплекс международных стандартов ИСО 13584 разработан в целях компьютерного представления данных, содержащихся в библиотеке деталей, а также для обмена этими данными. Целью настоящего стандарта является представление объективного способа, обеспечивающего передачу данных о библиотеках деталей и не зависящего от любого назначения системы, в которой эти данные будут использоваться. Благодаря характеру используемого описания оно может быть использовано не только для обмена файлами, содержащими детали, но и в качестве основы для реализации и совместного использования баз данных для элементов библиотеки деталей.

Комплекс международных стандартов ИСО 13584 выполнен в виде частей, каждая из которых публикуется отдельно. Части комплекса международных стандартов ИСО 13584 попадают в одну из следующих категорий: концептуальные описания, ресурсы программной реализации, методология описаний, проверка на соответствие, протокол просмотра и обмена данными и стандартизованное содержание. Данные категории описаны в ИСО 13584-1. Настоящий стандарт является одной из частей, посвященных методологии описаний.

Настоящий стандарт содержит правила и руководства для технических комитетов по стандартизации и поставщиков информации в целях создания онтологий продукции. Онтологии продукции состоят из иерархии характеристических классов деталей, выполненных в соответствии с общей методологией и обеспечивающих согласованность между поставщиками. Правила и руководства, приведенные в настоящем стандарте, содержат методы группировки деталей в характеристические классы деталей в целях образования иерархий; методы сопоставления свойств деталей с характеристическими классами деталей и со словарем элементов, содержащим классы и свойства деталей.

Данная часть ИСО 13584 нормативно ссылается на модель данных, описывающую процедуру обмена словарными данными. Настоящая спецификация языка EXPRESS разработана как общая модель ИСО 13584 и МЭК 61360. Она планируется к публикации как МЭК 61360-2. Для удобства указанная общая модель рассмотрена в настоящей части ИСО 13584 как справочное приложение, дублирующее нормативное содержание МЭК 61360-2. Настоящая часть ИСО 13584 также рассматривает отображения понятий, описанных здесь, на указанную общую модель. Для понимания приложения F, содержащего описание указанной модели, требуется знание языка EXPRESS. Язык EXPRESS определен в ИСО 10303-11:1994. Для понимания нормативных разделов настоящей части ИСО 13584 специальных знаний не требуется.

Настоящее второе издание настоящей части ИСО 13854 представляет нижеследующие возможности моделирования:

- возможность модельных ограничений свойств путем сужения области их значений;
- возможность моделирования и распознавания классов характеризации и классов категоризации;
- возможность моделирования агрегации и композиции с помощью одного механизма ресурсов;
- возможность описывать строки, имеющие внешние ссылки;
- возможность соединять классы, принадлежащие различным иерархиям классов.

В настоящем втором издании настоящей части ИСО 13584 удалены:

- возможность специализировать класс предметов как класс признаков, класс компонентов или класс материалов.

Примечание — Нижеследующие изменения гарантируют, что если словарь соответствует первому изданию настоящей части ИСО 13584, то он соответствует и настоящему изданию: (1) заменить **feature_class**, **component_class** и **material_class** на **item_class** по всему ссылочному словарю; (2) добавить каждому новому классу предметов **item_class** класс атрибутов реализаций совместного доступа **instance_sharable**, значение которых равно «true» для класса компонентов **component_class** и класса материалов **material_class**, и равно «false» для класса признаков **feature_class**; (3) добавить места для нумерации дополнительных атрибутов.

Настоящее второе издание ИСО 13584-42 представляет собой технический пересмотр ИСО 13584-42:1998, предусмотрительно сохраненного, чтобы обеспечить непрерывное использование и техническое обслуживание основанных на нем практических реализаций и удовлетворить нормативным ссылкам других частей ИСО 13584. Настоящее второе издание ИСО 13584-42 также включает Технический Корригендум ИСО 13584-42:1998/Cor.1:2003.

Комплекс ИСО 13584 имеет общее название «Системы промышленной автоматизации и интеграция. Библиотека деталей» и включает в себя следующие части:

- часть 1. Общие положения и фундаментальные принципы;
- часть 20. Логические ресурсы. Логическая модель выражений;

- часть 24. Логические ресурсы. Логическая модель библиотеки поставщика;
- часть 25. Логический ресурс: Логическая модель библиотеки поставщика с совокупными значениями и явным содержанием;
- часть 26. Логические ресурсы. Идентификация поставщика
- часть 31. Ресурсы программной реализации. Интерфейс геометрического программирования;
- часть 32. Ресурсы практической реализации: Язык OntoML: Язык разметки онтологии продукта;
- часть 35. Ресурсы практической реализации: Интерфейс в виде электронной таблицы для библиотеки деталей [Техническая спецификация];
- часть 42. Методология описания. Методология структурирования семейств деталей;
- часть 101. Протокол обмена и просмотра данных. Геометрический протокол обмена и просмотра данных с помощью параметрической программы;
- часть 102. Протокол обмена и просмотра данных: Протокол обмена и просмотра данных на основе спецификации соответствия ИСО 10303;
- часть 501. Ссылочный словарь для измерительных инструментов — Процедура регистрации;
- часть 511. Механические системы и компоненты общего пользования — Ссылочный словарь крепежных деталей.

Структура комплекса международных стандартов приведена в ИСО 13584-1. Нумерация частей комплекса отражает его структуру:

- части 10—19 устанавливают концептуальные описания;
- части 20—29 устанавливают логические ресурсы;
- части 30—39 устанавливают ресурсы программной реализации;
- части 40—49 устанавливают методологию описания;
- части 100—199 устанавливают требования к оформлению протоколов просмотра и обмена данными;
- части 500—599 устанавливают стандартизованное содержание.

Полный перечень стандартов комплекса ИСО 13584 приведен в сети Интернет:

gost.e-ecolog.ru

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы промышленной автоматизации и интеграция

БИБЛИОТЕКА ДЕТАЛЕЙ

Часть 42

Методология описания. Методология структурирования семейств деталей

Industrial automation systems and integration.
Parts library. Part 42. Description methodology. Methodology for structuring parts families

Дата введения — 2014—01—01

1 Область применения

Настоящий стандарт устанавливает принципы определения характеристических классов и свойств деталей, характеризующих деталь независимо от ее определения (идентификатора), данного поставщиком.

Правила и руководства настоящего стандарта являются обязательными для технических комитетов по стандартизации, ответственных за создание иерархии стандартных характеристик.

Данные правила рекомендуются для поставщиков и пользователей в качестве методологии построения их собственных иерархий.

Настоящий стандарт распространяется на:

- правила группировки деталей в листовые и нелистовые характеристические классы деталей;
- правила выбора надлежащих свойств, ассоциированных с характеристическим классом деталей;
- атрибуты, предоставляемые поставщиком информации, для описания характеристических классов и свойств деталей;
- механизмы соединения характеристических классов деталей с системами классификации;
- механизмы соединения характеристических классов, принадлежащих различным характеристическим иерархиям;
- спецификации сущностей и атрибутов информационной EXPRESS модели (модели на языке EXPRESS), обеспечивающие обмен указанными словарными данными;
- описание всех прочих объектов, отличных от деталей, характеризуемых своей принадлежностью к заданному классу и набором пар значений свойств. Соответствующая методология определена в настоящем стандарте.

Пример — Описание с помощью словаря, соответствующее данной части ИСО 13584, может использоваться для описания любого вида продукции в соответствии с Разделом 3.

Примечание 1 — Полная информационная EXPRESS модель обмена словарными данными, известная как общая словарная модель ИСО 13584/МЭК 61360, определена в ИСО 13584-25. Несколько уровней допустимой практической реализации общей словарной модели ИСО 13584/МЭК 61360, известной как классы соответствия, также определены в ИСО 13584-25. Класс соответствия 1 состоит из нескольких схем, установленных в настоящем стандарте (дублирующих информацию, содержащуюся в МЭК 61360-2), так же как и схема **ISO13584_IEC61360_dictionary_aggregated_extension_schema**, установленная в ИСО 13584-25 (дублируется в МЭК 61360-5). Более продвинутые классы соответствия, идентифицированные как классы соответствия 2, 3 и 4, установлены в ИСО 13584-25.

Настоящий стандарт не распространяется на:

- свойства, значения которых имеют комплексную структуру;

Примечание 2 — Информационная EXPRESS модель обмена свойствами, значения которых имеют комплексную структуру, определена в ИСО 13584-25.

- описания самих деталей;

Издание официальное

- описания функциональных моделей, которые могут ссылаться на некоторые классы деталей;
- описания таблиц, библиотек программ и документов, которые могут ссылаться на некоторые классы деталей;

Примечание 3 — Конструктивы EXPRESS ресурсов обмена указанными элементами информации определены в ИСО 13584-24:2003.

- описания систем управления библиотеками деталей.

Структура информации и методология, определенные в части «Методология описания» ИСО 13584, способствуют:

- интеграции в указанные архивы данных различных библиотек деталей, взятых от различных поставщиков информации, с механизмом постоянного доступа, обеспечиваемого словарем;
- ссылкам на другую библиотеку поставщика (доступную для получающей системы);
- ссылкам на иерархию стандартных характеристик (если такая иерархия существует);
- определению (конечным пользователем) иерархии локальной классификации (поисковой иерархии) и отображению данной иерархии в библиотеке поставщика, доступной для данной системы.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты, которые необходимо учитывать при использовании настоящего стандарта. В случае ссылок на документы, у которых указана дата утверждения, необходимо пользоваться только указанной редакцией. В случае, когда дата утверждения не приведена, следует пользоваться последней редакцией ссылочных документов, включая любые поправки и изменения к ним.

ИСО 8601:2004 Элементы данных и форматы для обмена информацией. Обмен информацией. Представление дат и времени (ISO 8601:2004 Data elements and interchange formats — Information interchange — Representation of dates and times)

ИСО 10303-11:2004 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS (ISO 10303-11:2004 Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual)

МЭК 61360-2:2012 Стандартные типы элементов данных с ассоциированной схемой классификации электрических компонентов— Часть 2: Схема словаря EXPRESS (IEC 61360-2 Standard data element types with associated classification scheme for electric components — Part 2: EXPRESS dictionary schema)

3 Термины и определения

В настоящем стандарте используются следующие термины и определения:

3.1 **абстрактный класс** (abstract class): Класс, члены которого являются также членами одного из его подклассов.

Примечание 1 — Абстрактные классы используются, если необходимо сгруппировать различные виды объектов в класс, включенный в иерархию.

Примечание 2 — В общей словарной модели ИСО 13584/МЭК 61360 могут определяться как абстрактные классификационные классы, так и абстрактные характеристические классы. Абстрактность — это только концептуальная характеристика класса. Данная характеристика в модели явно не представляется.

Примечание 3 — Через наследственность абстрактный характеристический класс допускает совместное использование, например, некоторых видимых свойств различными подклассами, соответствующими различным видам элементов.

3.2 **применимое свойство класса** (applicable property of a class): Применимое свойство является неотъемлемым свойством каждой детали, являющейся членом характеристического класса.

Примечание 1 — Каждая деталь, являющаяся членом характеристического класса, обладает аспектом, соответствующим каждому применимому свойству рассматриваемого характеристического класса.

Примечание 2 — Вышеуказанное определение является концептуальным. Нет требования, чтобы все применимые свойства класса использовались для описания каждой детали данного класса на уровне модели данных.

Примечание 3 — Все применимые свойства суперкласса также являются применимыми свойствами для подкласса рассматриваемого суперкласса.

Примечание 4 — Только свойства, определенные или унаследованные как видимые, а также импортированные свойства класса могут быть применимыми.

Примечание 5 — Для облегчения интеграции библиотек деталей и электронных каталогов, основанных на ИСО 13584-24:2003 и ИСО 13584-25, настоящий стандарт требует, чтобы только свойства, применимые для класса, использовались для описания их реализаций в библиотеках деталей и электронных каталогах.

3.3 **атрибут** (attribute): Элемент данных для компьютерного описания свойства, соотношения или класса.

Примечание — Атрибут описывает только одну подробность свойства, класса или соотношения.

Пример — Имя свойства, код класса, единица измерения, в которой представляются значения свойства.

3.4 **базовая семантическая единица** (basic semantic unit; BSU): Сущность, определяющая абсолютную, универсальную и уникальную идентификацию объекта домена приложений, представленную как элемент словаря.

Пример 1 — Словарь, соответствующий настоящему стандарту, способствует идентификации классов, свойств, источников информации и типов данных.

Пример 2 — Словарь, соответствующий ИСО 13584-24:2003, способствует идентификации классов, свойств, источников информации, типов данных, таблиц, документов и библиотек программ.

Пример 3 — В ИСО 13584-511 класс «болт с шестигранной головкой» идентифицирован как BSU. Свойство «класс допуска резьбы» также идентифицировано как BSU.

Примечание — Содержание базовой семантической единицы также представляется с помощью IRDI.

3.5 **характеристика продукции** (characteristic of a product, product characteristic): Неизменное *свойство,* характеристика *продукции,* значение которого фиксируется, как только продукция определена.

Примечание 1 — Изменение значения характеристики продукции приводит к изменению самой продукции.

Пример — Для шарикового подшипника внутренний и внешний диаметры являются характеристиками продукции.

Примечание 2 — См. ИСО 13584-24:2003, определение 3.12.

3.6 **класс** (class): Абстракция набора подобной продукции.

Примечание 1 — Продукция, соответствующая абстракции, определенной классом, называется членом класса.

Примечание 2 — Класс — особое понятие, которое может иметь самый широкий смысл в различных контекстах.

Пример — Продукция, используемая предприятием, и перечень существующей продукции, удовлетворяющий требованиям стандартов ИСО, — это два разных контекста. В этих двух контекстах (конкретное предприятие и стандарт ИСО) перечень продукции (например, члены класса одиночных шариковых подшипников) может быть разным: отдельное предприятие не может использовать сразу все виды одиночных шариковых подшипников.

Примечание 3 — Классы структурируются соотношениями включения в класс.

Примечание 4 — Класс продукции — это общее понятие по ИСО 1087-1. Правила, определенные в ИСО 704, рекомендуется использовать при определении обозначений и атрибутов класса продукции.

Примечание 5 — В контексте серии ИСО 13584 класс является либо характеристическим классом, ассоциированным со свойствами и используемым для характеристики продукции, либо классификационным классом, не ассоциированным со свойствами и не используемым для характеристики продукции.

3.7 **соотношение включения в класс** (class inclusion relationship): Соотношение между классами, формализующее условие включения в класс: если класс А является суперклассом для класса А1, то в любом контексте любой член А1 также является членом А.

Пример 1 — Перечень продукции, используемый на конкретном предприятии, и перечень всей продукции, удовлетворяющей требованиям стандартов ИСО, — это два примера различных контекстов.

Пример 2 — В любом контексте класс конденсаторов включает класс электролитических конденсаторов.

Примечание 1 — Соотношение включения в класс определяет иерархическую структуру классов.

Примечание 2 — Соотношение включения в класс — это концептуальное соотношение, не предписывающее что-либо на уровне представления данных. Оно не требует использования какой-либо базы данных или модели данных.

Примечание 3 — В модели, определенной в настоящем стандарте, «представительное» соотношение обеспечивает включение в класс. В настоящем стандарте рекомендуется использовать для включения в класс также и «условные» соотношения.

Примечание 4 — Соотношение включения в класс также называется отнесением к заданной категории.

3.8 **член класса** (class member): Продукт, соответствующий абстрактному понятию, определенному классом.

3.9 **свойство со значением, определенным классом** (class valued property): Свойство, имеющее одно значение для всего характеристического класса продукции.

Примечание 1 — Значение такого свойства (со значением, определенным классом) определяется не индивидуально для каждого отдельного продукта, а глобально для всего характеристического класса продукции.

Примечание 2 — Если все продукты характеристического класса имеют одинаковые значения заданного свойства, то определение данного свойства как свойства со значением, определенным классом, позволит избежать дублирования указанного значения при каждой реализации.

Примечание 3 — Свойства со значением, определенным классом, могут также использоваться для задания некоторой общности между различными характеристическими классами, если указанная общность не занята рассматриваемой иерархической структурой (см. Правило 4b).

3.10 **общая словарная модель ИСО 13584/МЭК 61360** (common ISO13584/IEC61360 dictionary model): Модель данных онтологии продукции, использующая язык моделирования информации EXPRESS и полученная в результате совместных усилий рабочих групп ИСО/ТК 184/ПК 4/РГ 2 и МЭК ПК3D.

Примечание 1 — Для общей словарной модели ИСО 13584/МЭК 61360 определены несколько уровней допустимой практической реализации (классы соответствия). Класс соответствия 1 состоит из нескольких схем, установленных в настоящем стандарте (дублирующая информация содержится в МЭК 61360-2). Схема **ISO13584_IEC61360_dictionary_aggregate_extension_schema** установлена в ИСО 13584-25 (дублируется в МЭК 61360-5). Прочие классы соответствия установлены в ИСО 13584-25 (это классы соответствия 2, 3 и 4).

Примечание 2 — В комплексе международных стандартов ИСО 13584 каждая заданная онтология продукции, адресованная заданной области продукции и основанная на общей словарной модели ИСО 13584/МЭК 61360, называется ссылочным словарем для указанной области.

3.11 **контекстно-зависимая характеристика продукции** (context dependent characteristic of product): Свойство *продукции,* значение которого зависит от некоторого *контекстного параметра.*

Примечание 1 — Для данной продукции контекстно-зависимая характеристика математически определяется как функция, область которой задается некоторым контекстным параметром, определяющим окружение продукции.

Пример — Для шарикового подшипника срок годности — это контекстно-зависимая характеристика, зависящая от радиальной нагрузки, осевой нагрузки и частоты вращения.

Примечание 2 — Адаптировано из ИСО 13584-24:2003, определение 3.22.

3.12 **контекстный параметр** (context parameter): Переменная, значение которой характеризует контекст, в который встроена *продукция.*

Пример 1 — Динамическая нагрузка на подшипник является контекстным параметром для данного подшипника.

Пример 2 — Температура среды, в которой измеряется сопротивление резистора, — это контекстный параметр для данного резистора.

Примечание 1 — Данное определение усиливает определение, установленное в ИСО 13584-24:2003: «это переменная, значение которой характеризует контекст, для которого предназначена продукция».

Примечание 2 — Значение свойства в ИСО 13584 представляется как тип элемента данных.

3.13 **тип элемента данных** (data element type): Блок данных, для которого имеются идентификация, описание и представление значения.

Примечание — В серии ИСО 13584 значение свойства представляется как тип элемента данных.

3.14 **словарные данные** (dictionary data): Набор данных, представляющих онтологию продукции, возможно, ассоциированную с классификацией продукции.

Примечание 1 — Словарные данные можно обменивать, используя некоторые классы соответствия общей словарной модели ИСО/МЭК.

Примечание 2 — Данное определение словарных данных усиливает предшествующее определение, взятое из первого издания настоящего стандарта: «Набор данных, описывающих иерархию характеристических классов продукции и свойств продукции».

3.15 **словарный элемент** (dictionary element): Набор атрибутов, составляющих словарное описание определенных объектов заданного домена приложений.

Пример 1 — Словарь, соответствующий настоящему стандарту, приводит описание классов, свойств, источников информации и типов данных.

Пример 2 — Словарь, соответствующий ИСО 13584-24:2003, приводит описание классов, свойств, источников информации, типов данных, таблиц, документов и библиотек программ.

3.16 **семейство продукции** (family of products): Перечень продукции, представляющий один характеристический класс.

Примечание — Данное определение усиливает определение, данное в ИСО 3584-24:2003: «Простое или групповое семейство деталей».

3.17 **особенность** (feature): Аспект продукции, который может быть установлен характеристическим классом и набором пар значений свойств.

Примечание 1 — В действительности особенность может быть только встроенной внутрь продукции, для которой она является аспектом.

Пример 1 — Головка винта — это особенность, описанная классом головок и рядом их свойств. Головка винта существует только вместе с самим винтом.

Примечание 2 — Особенности представляются классом элементов **item_class**, атрибут которых **instance_sharable** равен *false*.

Примечание 3 — Атрибут **instance_sharable** описывает концептуальный статус элемента: либо это отдельный элемент (значение **instance_sharable**=*true*), либо это особенность (тогда значение **instance_sharable**=*false*). Здесь не принимаются во внимание какие-либо ограничения на уровне представления данных в общей словарной модели ИСО 13584/МЭК 61360, представляющей несколько практических реализаций, совместно использующих одинаковое представление EXPRESS (одной или несколькими сущностями EXPRESS, зависящими от особенностей рассматриваемой практической реализации). Не существует механизма указания на возможность (невозможность) совместного использования значений данных реализации особенности.

Пример 2 — На одну и ту же реализацию класса головок винтов могут ссылаться несколько реализаций класса винтов. Это означает, что существует несколько головок винтов, но все они имеют один характеристический класс и одинаковый набор значений свойств. Атрибут instance_sharable показывает, что изменение данной реализации класса головок винтов приведет к изменению нескольких реализаций класса винтов.

3.18 **импортированное свойство** (imported property): Свойство, определенное в одном классе и выбранное другим классом (с тем же или другим ссылочным словарем с помощью условного соотношения), чтобы стать применимым в этом последнем классе.

Примечание 1 — Только свойства, видимые и/или применимые в классе, могут импортироваться из данного класса.

Примечание 2 — Импортирование между классами с различными ссылочными словарями допускает повторное использование свойств (определенных, например, в стандартных ссылочных словарях) без их повторного определения.

Примечание 3 — Импортирование между классами с одинаковыми ссылочными словарями допускает, что некоторые продукты могут выполнять несколько функций, требующих импортирования свойств из нескольких классов более высокого уровня.

Примечание 4 — Если свойство импортируется в новый класс, то оно сохраняет свой исходный идентификатор. Таким образом, соответствующие атрибуты не требуют дублирования.

Примечание 5 — Импортированное свойство является применимым для класса, в который оно импортировано.

3.19 **информация** (information): Факты, понятия или инструкции.

[ИСО 10303-1:1994, статья 3.2.20]

3.20 **информационная модель** (information model): Формальная модель ограниченного набора фактов, понятий или инструкций, удовлетворяющих установленным требованиям.

[ИСО 10303-1:1994, статья 3.2.21]

3.21 **поставщик информации, поставщик** (information supplier, supplier): Организация, определяющая онтологии или библиотеки поставщика в стандартном формате, установленном в настоящем стандарте, и несущая ответственность за их содержание.

Примечание — Данное определение поставщика усиливает определение поставщика информации, введенное в ИСО 13584-1:2001: «Организация, доставляющая библиотеки поставщика в стандартном формате, определенном в настоящем стандарте, и несущая ответственность за их содержание».

3.22 **международный идентификатор регистрации данных** (international registration data identifier): Международный уникальный идентификатор, определяющий домен приложений объекта в соответствии с ИСО/МЭК 11179-5.

Примечание 1 — Только международные идентификаторы регистрации данных, соответствующие ИСО/ТС 29002-5, могут использоваться в контексте серии ИСО 13584.

Примечание 2 — Международные идентификаторы регистрации данных могут использоваться для представления содержания базовой семантической единицы, идентифицирующей словарный элемент как строку.

Примечание 3 — Международный идентификатор регистрации данных может также использоваться для идентификации содержания атрибутов словарных элементов.

Пример — Единицы измерения свойств, значения (ограничения) свойств могут определяться идентификатором IRDI.

3.23 **представительное соотношение** (is-a relationship): Соотношение включения в класс, ассоциированное с наследственностью: если А1 *представляет* А, то каждый продукт, принадлежащий А1, принадлежит и А, и все описанное в контексте А автоматически дублируется в контексте А1.

Примечание 1 — Данный механизм обычно называют «наследственным».

Примечание 2 — В общей словарной модели ИСО 13584/МЭК 61360 представительное соотношение может определяться только между характеристическими классами. Рекомендуется определять только одну иерархию. Это обеспечивает наследование видимых и применимых свойств.

3.24 **условное соотношение, условный** (is-case-of relationship, case-of): Механизм импортирования свойств: если А1 связан *условно* с *А,* то определение продукции из А также покрывает продукцию из А1. Таким образом, А1 может импортировать любые свойства из А.

Примечание 1 — Целевые условные соотношения позволяют соединять вместе несколько иерархий соотношений включения в класс. При этом гарантируется, что ссылочные иерархии могут обновляться независимо.

Примечание 2 — Не существует ограничений по использованию условных соотношений для определения отдельных иерархий.

Примечание 3 — В общей словарной модели ИСО 13584/МЭК 61360 условное соотношение, в частности, может быть использовано в четырех случаях: (1) для связи характеристического класса и классификационного класса, (2) для импортирования (в контексте некоторых стандартных ссылочных словарей) некоторых свойств, ранее определенных в других стандартных ссылочных словарях, (3) для соединения ссылочного словаря пользователя с одним или несколькими стандартными ссылочными словарями, (4) для описания продукции с помощью свойств различных классов: если продукция класса А1 выполняет две различные функции и, таким образом, логически описывается свойствами, ассоциированными с двумя различными классами (А и В), то класс А1 может быть присоединен представительным соотношением к А и условным соотношением к В.

Примечание 4 — Конструктивы ресурсов языка EXPRESS для моделирования условных соотношений определены в приложении F настоящей части ИСО 13584.

3.25 **элемент** (item): Предмет, описываемый характеристическим классом, которому он принадлежит, и набором пар значений свойств.

Примечание 1 — Данное определение усиливает определение, данное в ИСО 13584-24:2003: «Предмет, описываемый структурой класса и набором свойств».

Примечание 2 — В комплексе международных стандартов ИСО 13584 продукция и особенности продукции, соответствующие составным свойствам, являются элементами.

3.26 **листовой характеристический класс** (leaf characterization class): Характеристический класс, который не специализируется далее в более точный характеристический класс.

Пример — Винты с потайной плоской головкой и крестообразным шлицем (тип Y) и винты с головной насадкой, с шестигранным углублением под ключ и метрической мелкой резьбой — это листовые характеристические классы, определенные в ИСО 13584-511.

3.27 **нелистовой характеристический класс** (non-leaf characterization class): Характеристический класс, который далее специализируется в более точный характеристический класс.

Пример — Компоненты с наружной резьбой и болты/винты с метрической резьбой — нелистовые характеристические классы, определенные в ИСО 13584-511.

3.28 **неколичественный тип элемента данных** (non-quantitative data element type): Тип элемента данных, идентифицирующий или описывающий объект с помощью кодов, аббревиатур, имен, ссылок или описаний.

3.29 **деталь** (part): Материал или функциональный продукт, являющийся деталью или компонентом определенного вида продукции.

[ИСО 13584-1:2001, определение 3.1.16]

3.30 **библиотека деталей** (parts library): Компьютерная онтология продукции и компьютерное описание перечня продукции, реализованное с помощью ссылок на онтологию.

Примечание — Настоящее определение усиливает определение, данное в первом издании настоящего стандарта: «Идентифицированный набор данных и, возможно, программ, генерирующих информацию о наборе деталей».

3.31 **продукция** (product): Предмет (вещество), изготовленный с помощью естественного (искусственного) процесса.

Примечание — В настоящем стандарте термин «продукция» понимается в самом широком смысле. Он включает устройства, системы, инсталляции, а также материалы, процессы, программное обеспечение и услуги.

3.32 **категоризация продукции, категоризация деталей, категоризация** (product categorization, part categorization, categorization): Рекурсивное разделение набора продукции на целевые подмножества.

Примечание 1 — Подмножества, указанные в категоризации продукции, называются классами категоризации продукции, или категориями продукции.

Примечание 2 — Категоризация продукции не является онтологией продукции. Она не может использоваться для категоризации продукции.

Примечание 3 — Никакие свойства не ассоциируются с категориями.

Примечание 4 — Несколько категорий одного набора продукции можно использовать в соответствии с их целевым назначением.

Пример — Категоризация UNSPSC (определенная ООН) является примером продукции, разработанной для анализа затрат.

Примечание 5 — С помощью условных соотношений несколько иерархий характеристического класса продукции можно соединить с иерархией категоризаций для создания единой структуры.

3.33 **класс категоризации продукции, класс категоризации деталей, класс категоризации** (product categorization class, part categorization class, categorization class): Класс продукции, составляющий элемент категоризации.

Пример — Компоненты производства и поставки и промышленная оптика — это примеры классов категоризации продукции, определенных UNSPSC.

Примечание 1 — В настоящем стандарте не приводятся правила выбора классов категоризации. Данное понятие введено: (1) для пояснения их отличия от характеристических классов, (2) для объяснения того, что одинаковые характеристические классы могут соединяться с любым количеством классов категоризации.

Примечание 2 — Никакие свойства не ассоциируются с классом категоризации.

3.34 **характеристика продукции, характеристика детали** (product characterization, part characterization): Описание продукции с помощью характеристического класса продукции, которому он принадлежит, и набора пар значений свойств.

Пример — Болт с шестигранной головкой по ИСО 4014 (Класс точности продукции = A, тип резьбы = M, длина = 50, диаметр = 8) — пример характеристики продукции.

3.35 **характеристический класс продукции, характеристический класс деталей, характеристический класс** (product characterization class, part characterization class, characterization class): Класс продукции, выполняющей одинаковые функции и имеющей общие свойства.

Примечание — Характеристический класс продукции может быть определен на различных уровнях детализации. Так формируется иерархия соотношений включения в класс.

Пример — Болт/винт с метрической резьбой и болт с шестигранной головкой — примеры характеристических классов продукции, определенных в ИСО 13584-511. Первый характеристический класс включается во второй. Транзистор и биполярный силовой транзистор — примеры характеристических классов продукции, определенных в МЭК 61360-4-DB. Второй класс включается в первый.

3.36 **онтология продукции, онтология детали, онтология** (product ontology, part ontology, ontology): Модель знаний о продукте, составленная с помощью формального и консенсуального представления понятий домена продукции в терминах идентифицированных характеристических классов, классовых соотношений и идентифицированных свойств.

Примечание 1 — Онтология продукции основана на реализации модели класса, позволяющей распознать и обозначить набор продукции (характеристический класс), имеющий подобные функции (например, *шариковый подшипник, конденсатор*). Онтология также позволяет выделить внутри класса различные подмножества продукции (реализации), которые считаются идентичными. Правила, определенные в ИСО 1087-1, рекомендуется использовать для формулировки обозначений и определений характеристических классов. Реализации определения не имеют. Они обозначаются классом (которому принадлежат) и набором пар значений свойств.

Примечание 2 — Онтологии связаны не со словами, а с понятиями, независимыми от какого-либо языка.

Примечание 3 — Термин «консенсуальный» означает, что концептуализация согласована в некотором сообществе.

Примечание 4 — Термин «формальный» означает, что онтология должна быть машинно-интерпретируемой. Некоторый уровень машинного мышления логически допустим в отношении онтологии, например, проверка соответствия, получение заключения.

Примечание 5 — Термин «идентифицированный» означает, что все классы и свойства, характеризующие онтологию, ассоциированы с глобально уникальным идентификатором, допускающим ссылку на данное понятие из любого контекста.

Примечание 6 — Модель данных онтологии, рекомендуемая в настоящем стандарте, является общей словарной моделью ИСО 13584/МЭК 61360. Ее простейшая версия установлена в настоящем стандарте. Более полные версии установлены в ИСО 13584-25 и МЭК 61360-5 (классы соответствия обоих документов — 1, 2, 3 и 4).

Примечание 7 — В настоящем стандарте каждая онтология продукции адресована заданному домену продукции, соответствующему общей словарной модели ИСО 13584/МЭК 61360, называемой ссылочным словарем для указанного домена.

Пример — Ссылочным словарем для электрических компонентов, определенных в МЭК 61360-4-DB, является онтология продукции для электрических компонентов, соответствующая общей словарной модели ИСО 13584/МЭК 61360. Она согласована всеми участниками МЭК/ПК3D. Корпоративный ссылочный словарь согласован экспертами, назначенными руководством от имени компании.

3.37 **свойство** (property): Определенный параметр, используемый для описания и дифференциации продукции.

Примечание 1 — Свойство описывает один аспект данного объекта.

Примечание 2 — Свойство определяется степенью общности его ассоциированных атрибутов. Типы и количество атрибутов, описывающих свойство с высокой точностью, установлены в настоящем стандарте.

Примечание 3 — Настоящий стандарт устанавливает три различных вида свойств: характеристики продукции, контекстные параметры и контекстно-зависимые характеристики продукции.

Примечание 4 — Указанное выше определение свойства усиливает предшествующее его определение, данное в первом издании настоящего стандарта: «Информация, которая может быть представлена типом элемента данных».

Примечание 5 — В комплексе международных стандартов значение свойства представлено как тип элемента данных.

3.38 **тип данных свойства** (property data type): Допустимый набор значений свойств.

3.39 **класс определений свойства** (property definition class): Характеристический класс продукции, в контексте которого определено свойство продукции.

Примечание — В общей словарной модели ИСО 13584/МЭК 61360 каждое свойство продукции имеет один класс определения его домена приложения. Свойство имеет смысл только для данного класса и всех его подклассов. Говорят, что свойство *видимо* в данном домене.

Пример — По ИСО 13584-511 гайка для высоты буртика является классом определений свойства. Аналогично, метрическая наружная резьба для основного диаметра наружной резьбы также является классом определений свойства.

3.40 **количественный тип элемента данных** (quantitative data element type): Тип элемента данных с численным значением, представляющим физическую величину, блок информации или набор объектов.

3.41 **ссылочный словарь** (reference dictionary): Онтология продукции, соответствующая общей словарной модели ИСО 13584/МЭК 61360.

Примечание — В комплексе международных стандартов ИСО 13584 онтология продукции, адресованная заданному домену продукции и основанная на общей словарной модели ИСО 13584/МЭК 61360, называется ссылочным словарем для указанного домена.

3.42 **конструктив ресурса** (resource construct): Набор сущностей языка EXPRESS, типов, функций, правил и ссылок, которые вместе составляют действительное описание данных.

[ИСО 13584-1:2001, статья 3.1.21]

Примечание — Данное определение адаптировано по отношению к определению конструктива ресурса, приведенному в ИСО 10303-1:1994: «Набор сущностей языка EXPRESS, типов, функций, правил и ссылок, которые вместе составляют действительное описание данных продукции».

3.43 **подкласс** (subclass): Класс, лежащий на ступень ниже другого класса в иерархии соотношений включения в класс.

Примечание — В общей словарной модели ИСО 13584/МЭК 61360, иерархии соотношений включения в класс определяются *представительными* соотношениями. Они также могут быть определены *условными* соотношениями.

3.44 **суперкласс** (superclass): Класс, лежащий на ступень выше другого класса в иерархии соотношений включения в класс.

Примечание 1 — В общей словарной модели ИСО 13584/МЭК 61360, иерархии соотношений включения в класс определяются представительными соотношениями. Они также могут быть определены условными соотношениями.

Примечание 2 — В общей словарной модели ИСО 13584/МЭК 61360 класс имеет, по крайней мере, один суперкласс, определенный с помощью представительного соотношения.

3.45 **библиотека поставщика** (supplier library): Библиотека деталей, по которой поставщик информации отличается от пользователя библиотеки.

Примечание — Данное определение усиливает определение, приведенное в ИСО 13584-1:2001: «Набор данных и, возможно, программ, для которых определяется поставщик, и которые описывают (в формате, установленном настоящим стандартом) набор продукции и/или набор представлений продукции».

3.46 **видимое свойство** (visible property): Свойство, определение которого имеет смысл в области применения данного характеристического класса, но которое не обязательно применяется к продукции, принадлежащей данному классу.

Примечание 1 — Понятие «имеет смысл в области применения данного характеристического класса» означает, что наблюдатель может определить (для любого характеристического класса продукции), применимо ли данное свойство и (если оно применимо) какому аспекту продукции оно соответствует.

Примечание 2 — Понятие «видимое свойство» допускает совместное использование определения свойства характеристическими классами продукции, где применение данного свойства не обязательно.

Пример — Свойство — нерезьбовая поверхность — имеет смысл для любого класса винтов. Однако оно применяется только для винтов, имеющих гладкую нерезьбовую часть. Данное свойство может определяться как видимое на уровне винта. Оно становится применимым только в некотором подклассе.

Примечание 3 — Все видимые свойства суперкласса, являющегося характеристическим классом продукции, также являются видимыми для его подкласса.

Примечание 4 — Для интеграции библиотек компонентов и электронных каталогов, определенных в ИСО 13584-24:2003, ИСО 13584-25 и настоящем стандарте, необходимо, чтобы только свойства, применимые для класса, использовались для характеристики их реализаций в библиотеках деталей и электронных каталогах.

Примечание 5 — Данное определение видимого свойства усиливает предшествующее определение, приведенное в ИСО 13584-24:2003: «Свойство, определенное для некоторого класса продукции, которое не обязательно применять для различных видов продукции данного класса».

4 Аббревиатуры

BSU — Базовая семантическая единица (Basic Semantic Unit);

DET — Тип элемента данных (Data Element Type);

ICS — Международная классификация стандартов (International Classification of Standards);

IRDI — Международный идентификатор регистрации данных (International Registration Data Identifier);

MathML — Схема XML математических обозначений (XML schema for mathematical notations);

SI — Международная система единиц СИ (Système International d'Unités (International System of Units));

UNSPSC — Код ООН стандартных продукции и услуг (United Nations Standard Products and Services Code).

5 Описание иерархии характеристических классов продукции

5.1 Соотношения между категориями продукции и онтологиями продукции

В настоящем стандарте определены конструктивы ресурсов разработки двух видов иерархий продукции:

- категории продукции, когда набор продукции рекурсивно разбивается на подмножества для организационных целей (такие иерархии не ассоциируются со свойствами);
- онтологии продукции, когда характеристические классы продукции специализируются в более точных характеристических классах (такие классы ассоциируются со свойствами).

Категории продукции никогда не используются для характеристики продукции. Поэтому изменения в категории продукции не влияют на характеристики продукции, указанные в структуре продукции или его базе данных. С другой стороны, изменения в онтологиях продукции оказывают влияние на все существующие характеристики продукции. Категории продукции и онтологии продукции имеют различный жизненный цикл. Онтологии продукции устойчивы, тогда как категории продукции могут изменяться во времени, в пространстве и по видам пользователей без каких-либо последствий для характеристик продукции.

Как следствие, категории продукции и онтологии продукции нельзя смешивать и нельзя соединять представительными соотношениями. Если оказывается возможным объединить несколько онтологий в одну структуру (в несколько поисковых структур), то характеристические классы можно соединить с классами категоризации, использующими условные соотношения.

5.2 Соотношения между классами

5.2.1 Соотношение включения в класс

В объектно-ориентированном анализе соотношение включения в класс (отнесения к заданной категории) составляет базовую операцию моделирования области знаний. Класс A1 называют родовым для класса A, если в любом контексте каждый объект, принадлежащий классу A1, также принадлежит и классу A.

В большинстве объектно-ориентированных языков соотношение включения в класс представляется наследственностью: если класс A является родовым для класса A1, то не только все реализации A1 принадлежат A, но также и все свойства, определенные для A, также применимы в A1. Более того, данный механизм наследственности является только механизмом, позволяющим двум классам совместно использовать одни и те же свойства. Рассматриваемая идентификация соотношений включения в класс и наследственности, часто ведущая к множественной наследственности, делает объектно-ориентированную модель немодульной, затрудняет ее использование в контексте, мало отличающемся от исходного. Для обеспечения модульности онтологии продукции, для модели онтологии, определенной в ИСО 13584, определены два различных представления соотношений включения в класс:

- обычное представительное соотношение, определенное в настоящем стандарте, позволяет отдельным группам или организациям создавать конкретные иерархии, отнесенные к заданной категории, или онтологии (категоризации) продукции с простым наследованием свойств;

- условное соотношение с конструктивами ресурсов EXPRESS обеспечивает модульность онтологий. Оно не означает автоматического наследования. Оно позволяет родовым классам явно импортировать некоторые (или все) видимые и/или применимые свойства для класса, отнесенного к некоторой категории. Данное соотношение позволяет: (1) устанавливать связь характеристического класса с одним или несколькими классами категоризации, (2) импортировать в рассматриваемый контекст некоторых иерархий стандартных характеристик некоторые свойства, ранее определенные в других иерархиях стандартных характеристик, (3) соединять словарь пользователя с одной или несколькими иерархиями стандартных характеристик, (4) описывать продукт, используя свойства различных классов. Если класс продукции А1 выполняет две различные функции и, таким образом, описывается свойствами, ассоциированными с двумя различными классами (например, А и В), то класс А1 может быть соединен представительным соотношением с А и условным соотношением с В.

Отметим, что соотношение включения в класс может использоваться как для группировки нескольких подклассов, представляющих различные виды объектов, внутри общего абстрактного суперкласса, так и для определения порядка перевода нескольких специализаций видов объектов, определенных заданным суперклассом, в специализации более высокого уровня, определенные специализированными подклассами. В обоих случаях представительное соотношение используется с учетом наследственности. Условное соотношение используется без учета наследственности.

5.2.2 Агрегация и композиция

Одни виды продукции могут быть собраны из других видов продукции. Например, болтовая сборка может включать болты, шайбы и гайки. Такие соотношения называют *агрегациями.*

Некоторые виды продукции трудно характеризовать простыми свойствами. Их легче описывать несколькими особенностями. Каждая особенность описывается как с помощью ссылки на характеристический класс, так и парой значений свойств. Указанное соотношение между характеристическим классом и всеми классами особенностей называют *композиционным* соотношением.

Пример 1 — Винт можно характеризовать его головкой (например, винт, принадлежащий классу шестигранных головок, можно описать высотой головки и расстоянием между гранями), резьбой (принадлежащей, например, классу метрических резьб) и способом закручивания. Соотношение между классом винта и всеми классами его особенностей — композиционное соотношение.

Существует принципиальное отличие между агрегацией и композицией. Агрегация применяется, если один продукт, включенный в другой продукт, может существовать независимо от него. Срок службы сборки может отличаться от срока службы ее компонентов деталей. Иногда компоненты меняют без контроля всей сборки. И наоборот, композиция применяется, если жизненный цикл продукции равен жизненному циклу составляющих компонентов. Данные компоненты не могут существовать без продукции, куда они встроены. Компоненты не могут изменяться без изменения всего вида продукции, к которому они относятся. Таким образом, отличие между композицией и агрегацией, а также между компонентом и отдельным продуктом четко определяется на концептуальном уровне.

Отметим, что указанное концептуальное отличие не обязательно отражается на уровне представления данных.

Пример 2 — Резьба винта, удовлетворяющая стандартам ИСО, может существовать только как деталь винта. Таким образом, на концептуальном уровне она считается неотъемлемым свойством винта (композицией). Тем не менее, резьбы винтов, удовлетворяющие стандартам ИСО, существуют только для ограниченного набора их размеров. Поэтому для эффективного хранения и обмена данных необходимо:

- рассматривать резьбы винтов, удовлетворяющие стандартам ИСО, как набор предварительно определенных экземпляров, которые можно хранить, например, в таблице,

- обменивать данный набор экземпляров независимо от каких-либо винтов,

- описывать винт без указания значений свойств его резьбы, просто ссылаясь на некоторую информацию по резьбе, обмениваемую отдельно.

В общей словарной модели ИСО 13584/МЭК 61360 понятия «композиция» и «агрегация» считаются концептуально отличными без указания требований к представлению данных. Как продукция, так и ее особенности должны моделироваться как классы элементов **item_class.** Композиционные и агрегационные соотношения представляются свойствами. Концептуальное отличие фиксируется заданием булевского атрибута **item_class,** называемого **instance_sharable** (совместно используемый экземпляр). Если значение атрибута **instance_sharable** класса элементов **item_class** равно *true,* то представлен отдельный продукт, существующий независимо от всех агрегационных соотношений. На эти практические экземпляры указанного класса могут производиться ссылки любого числа других практических экземпляров. Если значение атрибута **instance_sharable** класса элементов **item_class** равно *false,* то это класс особенностей, представляющих составное свойство. Практический экземпляр такого

класса может существовать, если он встроен только в один продукт. От данного экземпляра зависит выбор между альтернативами: несколько практических экземпляров особенностей, моделируемых одинаковым набором пар значений свойств, представляются 1) несколькими блоками данных EXPRESS или 2) одним и тем же блоком данных в файле обмена данными.

5.3 Одновременное описание характеристических классов продукции и свойств продукции

Характеристическая иерархия должна формироваться как одна или несколько древовидных структур с единой наследственностью. На каждом уровне иерархии подклассы характеристических классов продукции должны быть, по возможности, взаимно исключаемыми.

- корнем каждой иерархии должен быть характеристический класс продукции, включающий всю продукцию, определяемую характеристической иерархией;
- ниже корня каждый нелистовой характеристический класс должен разбиваться на нелистовые характеристические подклассы до получения листовых характеристических классов продукции. При выборе указанных характеристических классов необходимо выполнять определенные правила (Правило 3). Каждый из данных характеристических классов можно использовать для характеристики продукции в целом.

Часто рекомендуется начинать с листовых характеристических классов, соответствующих тем семействам продукции, которые в данный момент широко используются в рассматриваемом домене. При этом для каждого класса все экземпляры могут быть описаны одними и теми же свойствами. Нелистовые характеристические классы создаются прогрессивно над указанными классами с помощью критерия (правила) конкретизации, определенного в разделе 6.1.3.

Как иерархии стандартных характеристик, так и любые заданные иерархии поставщика должны одновременно определять:

- (иерархические) характеристические классы продукции;
- (неиерархические) свойства.

Смысл каждого характеристического класса продукции и каждого свойства должен быть таким, чтобы для данной продукции пользователь имел возможность определить:

- характеристические классы, которым данный продукт принадлежит, и которым он не принадлежит;
- аспект продукции, которому соответствует каждое применимое свойство.

Одновременное определение характеристических классов продукции и свойств, связанных с соответствующим характеристическим классом, улучшает определение продукции. Охват характеристического класса продукции становится яснее по свойствам, применимым к рассматриваемому продукту. Смысл свойства поясняется характеристическим классом продукции, определяя область его возможных приложений.

С помощью определения характеристических классов продукции через иерархию с наследственностью, определение каждого свойства может быть сведено к наивысшему уровню характеристического класса, для которого данное свойство является существенным и применимым в нескольких подклассах.

5.4 Применимые и видимые свойства

Для определения иерархии точной характеристики необходимо удовлетворить двум требованиям:

- точное определение свойства требует ссылки (в его определении) на вид продукции, для которых данное свойство является существенным;
- точное определение класса требует использования критерия членства в данном классе.

В комплексе международных стандартов ИСО 13584 данные два требования представляются с помощью двух соотношений «класс-свойство»:

- область, для которой свойство является существенным, устанавливается с помощью класса, составляющего *класс определений свойств.* Текстовое определение свойства содержит (в общем случае) ссылки на данный класс. Таким образом, только определенная продукция, являющаяся членом данного класса определений свойств (или любого класса, включенного в данный класс), может иметь аспект, отображаемый однозначно в рассматриваемое свойство. Указанное свойство называется *видимым* (т. е. существенным) для данного класса и всех его подклассов.

Пример 1 — Особенности замены смазки, определенные внутри класса подшипников (например, устройство подвода смазки к поверхности качения или поверхности скольжения подшипника), четко прописываются для любого подшипника. Однако данные особенности не являются однозначными при характеристике любой продукции, не являющейся подшипником.

- критерии членства в классе устанавливаются с помощью указания набора свойств. Только продукция, имеющая аспект, соответствующий каждому из указанных свойств, является членом рассматриваемого класса. Все свойства, являющиеся критериями членства в классе, называются *применимыми* в данном классе.

Пример 2 — Внутренний диаметр и наружный диаметр являются свойствами подшипника качения. Продукт, для которого данные характеристики не существуют, не является подшипником качения.

Если вся продукция подкласса входит в суперкласс, то применимость наследуется через иерархию соотношений включения в класс.

Пример 3 — Внутренний диаметр и наружный диаметр — это свойства, обязательные для всех видов продукции, принадлежащей к одному из подклассов класса подшипников качения (например, шариковым подшипникам и роликовым подшипникам).

Если свойство *P* определено вместе с классом *C* (как классом определений свойств), то может оказаться, что некоторые продукты в *C* не имеют аспектов, соответствующих *P*. Таким образом, не все видимые свойства класса обязательно являются применимыми свойствами для данного класса. По этой причине видимые свойства и применимые свойства нужно отличать друг от друга.

Пример 4 — Особенности замены смазки не являются существенными для герметичных подшипников, которые смазываются только один раз и навсегда. Тем не менее, герметичные подшипники также относятся к классу подшипников.

Если определение свойства содержит спецификацию класса определений свойств, то для корня дерева данное свойство должно быть видимым. Применимые свойства определяются, если определен характеристический класс. Существует два механизма создания применимых свойств в характеристическом классе:

- использование (через наследственность) свойства (в качестве применимого), определенного ранее как видимое в данном классе;
- импортирование свойства (принадлежащего другой иерархии) из другого класса с помощью условного соотношения «класс-класс». Если данное условное соотношение отражает тот факт, что определение ссылочного класса распространяется на продукцию ссылочного класса, то свойства данного ссылочного класса являются существенными для продукции ссылочного класса.

Примечание 1 — Основной целью использования условного соотношения является совместное использование свойств различными иерархиями.

Примечание 2 — Для интеграции библиотек деталей и электронных каталогов, соответствующих ИСО 13584-24:2003 и ИСО 13584-25, вышеуказанные международные стандарты содержат требования, чтобы свойства, применимые к некоторому классу, использовались для характеристики их реализаций в указанных библиотеках деталей и электронных каталогах.

5.5 Назначение иерархии стандартных характеристик

Реальный мир бесконечно разнообразен. Поэтому характеристические классы продукции и свойства, составляющие иерархию стандартных характеристик, не могут исчерпывающим образом описать каждый продукт. Они предназначены только для характеристики нескольких доменов продукции с помощью тех характеристических классов и тех свойств, для которых достигнуто согласие. Среди прочих возможностей иерархии стандартных характеристик нужны для:

- характеристики поставщиков нескольких видов продукции;
- обеспечения взаимозаменяемости различной продукции;
- поиска поставщиков нескольких видов продукции.

5.6 Использование иерархии стандартных характеристик

Для описания своих собственных видов продукции, пользователь комплекса международных стандартов ИСО 13584 может:

- делать ссылки на стандартные характеристические классы продукции и свойств, если они определены иерархией стандартных характеристик;
- определить свою собственную иерархию характеристик и соединить ее с одной или несколькими иерархиями стандартных характеристик с помощью условных соотношений, импортировав из нее все свойства, существенные в рассматриваемом контексте.

В последнем случае пользователь должен соединить каждый свой собственный характеристический класс продукции с нижним характеристическим классом иерархии стандартных характеристик, включающим всю продукцию рассматриваемого характеристического класса. Далее пользователь (при необходимости) может уточнить свой характеристический класс с помощью привлечения других

более специальных классов. Чем выше уровень соединения, тем меньше пользы можно извлечь из указанной иерархии стандартных характеристик и соответствующих свойств.

Первое приближение выполнить просто. Второе приближение позволяет построить иерархию, более адаптированную к конкретным потребностям конкретного пользователя. В любом случае (благодаря применению механизма импортирования, определенного условным соотношением) автоматическая интеграция данных из различных источников может быть выполнена для всех данных, содержащих ссылки на стандартные свойства.

5.7 Свойства со значением, определенным классом

Перспективы использования реальных объектов бесконечно разнообразны. Поэтому имеется значительное количество различных структур нелистовых характеристических классов продукции (см. пример в разделе 6.1.4).

В данном случае для иерархии стандартных характеристик выбирается структура, обеспечивающая максимальную применимость факторизованных свойств. Каждая новая рассматриваемая перспектива представляется как свойство, которому можно назначить только одно значение, соответствующее характеристическому классу дерева. Назначение свойства для характеристического класса определяется на любом уровне поддерева. Значение, назначенное таким способом, унаследует все подклассы.

Тип данных такого свойства представляется как набор уникальных кодов. Каждый код ассоциируется со своими читабельными и переводимыми представлениями.

Запрос конкретных значений данного свойства на уровне некоторых нелистовых характеристических классов позволит получить наборы подклассов данного класса независимо от иерархической декомпозиции данного класса.

5.8 Совместимость ИСО 13584 и МЭК 61360

Схемотехника, представленная в настоящем стандарте, разработана совместно рабочими группами ИСО/ТС 184/ПК 4/РГ 2 и МЭК/ПК 3D. Она объединяет требования пользователей МЭК 61360 и ИСО 13584 и может быть расширена с помощью составления общей словарной модели ИСО 13584/МЭК 61360, схемотехника которой является простейшим функциональным подмножеством. Данное подмножество установлено как в МЭК 61360-2, так и в настоящем стандарте. Ее можно использовать как в контексте ИСО 13584, так и в контексте МЭК 61360 для обмена простыми словарными данными.

Схемотехника, представленная в настоящем стандарте, включает ряд сущностей, определенных как абстрактные супертипы без каких-либо подтипов. Данные ресурсы — это механизмы, обеспечивающие указанные расширения. В соответствии с ИСО 10303-11:1994 сущности, определенные как абстрактные супертипы, непосредственно использовать нельзя.

Два атрибута, определенные схемой, установленной в настоящем стандарте, относятся конкретно к требованиям, определенным МЭК 61360. Рассмотрены расширения:

a) в сущности **item_class** — это атрибут **value_code** по выбору;

b) в сущности **value_domain** — это атрибут **terms** по выбору.

Данные атрибуты по выбору не рекомендуется использовать для пользователей стандартов ИСО. Данные атрибуты являются частью общей словарной модели ИСО 13584/МЭК 61360. Они перерабатываются в процессе практической реализации. При этом необходимо обеспечить соответствие обоим стандартам, чтобы добиться полной совместимости словарных данных ИСО 13584 и МЭК 61360.

6 Правила создания иерархий характеристических классов продукции

Для упрощения анализа при создании иерархий стандартных характеристик (иерархий поставщиков) необходимо руководствоваться нижеследующими правилами, позволяющими разбить анализ на две фазы. Первый набор правил относится к выбору иерархии характеристических классов. Второй набор правил рассматривает ассоциацию свойств с характеристическим классом продукции. Фактически — это итерационный процесс. Идентификация новых свойств ведет к пересмотру некоторых структурных решений, принятых при разработке иерархии классов, и наоборот.

6.1 Выбор иерархии характеристических классов

6.1.1 Область приложения

Правило 1 — Область, охватываемая рассматриваемой иерархией.

Определение корневого класса характеристической иерархии должно четко идентифицировать все виды продукции, характеризуемые при использовании данной характеристической иерархии.

Рекомендуется разработать иерархии стандартных характеристик для каждого домена продукции, ссылающегося на стандарты ИСО и МЭК.

Корни указанной иерархии (и некоторые другие узловые точки) могут соединяться с Международной классификацией стандартов (ICS) для описания домена, охватываемого рассматриваемой иерархией.

Примечание — Соединение с ICS и, возможно, с другими категоризациями, может быть формально представлено условным соотношением, определенным в приложении F настоящей части ИСО 13584.

6.1.2 Верхнее сечение иерархии классов

Правило 2 — Организация области продукции.

Роль верхнего сечения иерархии стандартных характеристик — организация нескольких аспектов домена продукции с помощью соотношений включения в класс и факторизация рассматриваемых свойств. Это необходимо для определения абстрактных классов, содержащих элементы различной природы (классов продукции и классов особенностей). На данном уровне некоторые видимые свойства используются совместно. Такие классы называются абстрактными, так как все члены указанных классов также обязательно являются членами некоторых их подклассов.

Если нет необходимости в совместном использовании некоторых свойств различных по природе элементов (классов продукции и классов особенностей), то могут быть определены отдельные иерархии.

6.1.3 Нижнее сечение иерархии классов

Правило 3 — Правило конкретизации.

Ниже верхнего сечения иерархии можно создавать только характеристические классы продукции:

- если допускается (рекомендуется) заказывать (искать) продукт в виде члена указанного характеристического класса;

- если пользователь может обоснованно выбрать продукт из характеристического класса, чтобы представить существенное состояние (фазу) рассматриваемого процесса проектирования.

Пример — Использование одной группы винтов для металла, дерева или листового металла в рамках одного характеристического класса «винт» противоречит правилу «конкретизации». В процессе проектирования инженер никогда не выберет винт, не зная, куда он вкручивается: в металлическую или деревянную поверхность. Следовательно, под верхним сечением класса «болты/винты/шпильки» необходимо определить другие характеристические классы продукции для различных типов винтов.

6.1.4 Возможные перспективы иерархии классов

В некоторых точках иерархии классов могут быть использованы несколько точек зрения для определения подструктуры рассматриваемой иерархии.

Правило 4a — Максимальная применимость.

Если приложение правила конкретизации (Правило 3) ведет к появлению различных допустимых структур для ряда характеристических классов (в зависимости от выбранной точки зрения), то следует выбрать структуру, обеспечивающую максимальную применимость факторизованных свойств.

Правило 4b — Свойства со значениями, определенными классом.

Другие существенные точки зрения, не выбранные по правилу максимальной применимости (Правило 4a), представляются свойствами со значением, определенным классом (раздел 5.7). Их пользователь запрашивает, а затем выбирает множество характеристических классов, соответствующих другим точкам зрения.

Правило 4c — Уровень со значением, определенным классом.

Значение, определенное классом, задается на уровне характеристического класса, где данное значение свойства применяется для *каждого* листового характеристического класса соответствующего поддерева. Это значение не должно применяться для ближайшего родительского характеристического класса.

Пример — Характеристический класс круглых подшипников можно разбить на шариковые, игольчатые и конические подшипники, а также на герметичные и негерметичные подшипники. Первая перспектива выбирается в целях структуризации (Правило 4a). Свойство (со значением, определенным классом), имеющее название «is_sealed» (герметичный) и принимающее постоянное БУЛЕВО значение в нижнем характеристическом классе, определяется как свойство (со значением, определенным классом) на уровне характеристического класса круглых подшипников (Правило 4b). Данному свойству назначается значение во всех классах, где все подшипники либо герметичные (is_sealed = TRUE) либо негерметичные (is_sealed = FALSE).

6.2 Ассоциация свойств

6.2.1 Описания свойств

Правило 5 — Выбор свойств.

Как минимум, свойства, характеризующие нелистовые характеристические классы продукции и используемые для поиска продукции для подклассов указанного нелистового характеристического класса, должны ассоциироваться как «видимые» или как «видимые и применимые» в соответствии с иерархией стандартных характеристик.

Свойства, не используемые (редко используемые) для поиска, могут добавляться по ситуации.

6.2.2 Семантическая идентификация свойств

Нижеследующее правило дает два критерия принятия решения, если два свойства имеют одну и ту же семантику.

Правило 6 — Семантическая идентификация.

Два характеристических свойства двух различных видов продукции могут факторизоваться на более высоком уровне иерархии, если и только если они имеют один и тот же семантический смысл. Принятое решение обосновывается с помощью одного из двух нижеследующих критериев:

- критерий взаимозаменяемости: два вида продукции могут, в некоторых обстоятельствах, быть взаимозаменяемыми. После взаимного обмена оба характеристических свойства должны иметь идентичные значения;

- критерий гомогенности обработки деталей: два свойства играют идентичные роли по отношению к некоторому процессу обработки (автоматическому или неавтоматическому) продукции.

В противном случае необходимо определить два различных свойства.

Пример 1 — Свойства «диаметр резьбы винтов с шестигранной головкой» и «диаметр резьбы винтов с цилиндрической головкой» удовлетворяют критерию взаимозаменяемости.

Пример 2 — Свойства «масса» (расчетная суммарная масса детали), «обозначение» (нужное для составления списков деталей), «диаметр конических хвостовиков инструментов» (для автоматов замены инструмента), «наружный диаметр цилиндрических электронных компонентов» (для автоматического размещения деталей) удовлетворяют критерию гомогенности при автоматической обработке деталей.

Пример 3 — Диаметр резьбы винтов для дерева или винтов для металла — это характеристические свойства, не являющиеся одним и тем же свойством.

6.2.3 Правило факторизации

Невозможно в каждом отдельном случае определять иерархию, на которой:

- каждое свойство, имеющее одну и ту же семантику в нескольких подклассах, факторизуется (то есть определяется только однажды и наследуется) как уникальное свойство на уровне восходящего характеристического класса;

- любое свойство, определенное на уровне листового характеристического класса продукции, применяется фактически (то есть имеет значение) для каждого вида продукции в каждом подклассе.

При создании промежуточного характеристического класса свойства защищаются от факторизации правилом конкретизации (Правило 3), используемым для определения иерархии характеристических классов.

Нижеследующее правило определяет порядок систематической факторизации свойств с одной и той же семантикой при условии выполнения правила конкретизации.

Правило 7 — Применимость унаследованных свойств.

Два свойства с одной и той же семантикой (Правило 6) в двух характеристических классах факторизуются как уникальные свойства, определенные на уровне общего восходящего характеристического класса. Если данное свойство неприменимо к некоторым подклассам, то оно должно определяться как «видимое». Данное свойство может быть описано как «применимое» в нескольких подклассах, где оно является «видимым». Если свойство описано как применимое, то его определение должно быть таким, что для любого подкласса нет сомнений в его применимости. При его применении не должно быть сомнений в аспекте продукции, которому данное свойство соответствует.

Пример 1 — Свойство «материал», то есть «материал, из которого сделан продукт», не может быть факторизовано, так как не указано, как данный материал соотносится со свойствами, например, токарного резца с напаяной твердосплавной кромкой.

Пример 2 — Свойство «однокомпонентный материал», то есть «свойство, применимое к продукции только из однокомпонентного материала (значением данного свойства является код рассматриваемого однокомпонентного материала)" соответствует правилу семантической идентификации (Правило 7).

7 Словарные элементы, описывающие свойства продукции

Свойства продукции подразделяются на:
- характеристики продукции;
- контекстно-зависимые характеристики;
- контекстные параметры.

7.1 Отображение свойств на общую словарную модель ИСО 13584/МЭК 61360

В общей словарной модели ИСО 13584/МЭК 61360 свойство отображается на сущность **property_ BSU,** имеющую свою идентификацию, и на сущность **property_DET,** имеющую описание. Отметим, что сущность **property_DET** абстрактна, поэтому нужно использовать ее подтипы.

Характеристика продукции отображается на независимый подтип **non_dependent_P_DET** свойства **property_DET.** Контекстно-зависимая характеристика отображается на зависимый подтип **dependent_P_DET** свойства **property_DET**. Контекстный параметр отображается на условный подтип **condition_DET** свойства **property_DET**.

Характеристический класс, определяющий видимость свойства, отображается на атрибут **property_BSU.name_domain**. Применимость свойства определяется классами, к которым данное свойство применимо посредством атрибута **class.described_by** данных классов.

В настоящем разделе описаны основные атрибуты, непосредственно или косвенно связанные со свойствами.

7.2 Атрибуты

Каждое свойство продукции описывается словарным элементом, содержащим следующие атрибуты:
- код;
- класс определений;
- тип данных;
- предпочтительное имя;
- сокращенное имя;
- предпочтительное буквенное обозначение;
- синонимические буквенные обозначения;
- синонимическое имя;
- классификация типов свойств;
- определение;
- исходный документ для определения;
- примечание;
- замечание;
- единица измерения;
- условие;
- формула;
- формат значения;
- дата исходного определения;
- дата текущей версии;
- дата текущего пересмотра;
- № версии;
- № пересмотра;
- данные, не рекомендуемые для использования;
- интерпретация нецелесообразности использования данных;
- административные данные.

Запись каждого атрибута структурируется следующим образом (записи могут быть опущены, если атрибут является неприменимым):
- Имя атрибута;
- Obj**:** Цель использования;
- Descr**:** Описание;
- Oblig: Обязательство;
- Trans: Необходимость перевода на другой язык;
- For: Формат представления (максимальное число и тип символов в строке);

- Mapp: Отображение на атрибуты используется в общей словарной модели ИСО 13584/МЭК 61360;
- Expl: Пример.

Примечание — Для внутренних целей внутри компании могут быть определены другие атрибуты (коды атрибутов), которые, однако, не могут использоваться для обмена.

7.2.1 Код

Obj: Идентифицировать свойства характеристического класса продукции. Дать абсолютную идентификацию внутри рассматриваемого словаря данных, ассоциируемого с кодом поставщика информации и № версии.

Descr: Базовая семантическая единица, ассоциируемая со свойством.

Oblig: Обязательный.

Trans: Перевода не требуется.

Mapp: Отображение **property_DET\dictionary_element.identified_by\basic_semantic_unit.code**

7.2.2 Класс определений

Obj: Указать, для какого характеристического класса определено данное свойство.

Descr: Код характеристического класса, являющегося корнем дерева, где данное свойство является видимым.

Oblig: Обязательный.

Trans: Перевода не требуется.

Mapp: Отображение **property_DET\dictionary_element.identified_by\property_BSU.name_domain**

7.2.3 Тип данных

Obj: Описать тип данных свойства. Этот тип данных описывает набор значений, которые могут быть назначены свойству.

Descr: Тип данных, соответствующий системе типов, установленной в общей словарной модели ИСО 13584/МЭК 61360 и описывающей типы данных свойств.

Oblig: Обязательный.

Trans: Перевода не требуется.

Mapp: Отображение **property_DET.domain**

Примечание — Отображение **property_DET.domain** ссылается на тип данных **data_type,** относимый к подтипу для различных допустимых типов данных. Таким образом, тип данных непосредственно прикрепляется к свойству. Однако механизм идентификации, использующий поименованный тип **named_type** и тип данных семантического блока **data_type_BSU,** также доступен для случаев, когда один и тот же тип данных предназначен для различных свойств. Если тип данных **data_type** предназначен для различных свойств, то определение рассматриваемого типа **data_type,** как поименованного типа **named_type,** способствует выборочному обновлению рассматриваемого типа данных и соответствующих ему свойств.

7.2.4 Предпочтительное имя

Obj: Задать имя свойству (полной длины, по возможности). Используется для связи и облегчения понимания.

Descr: Предпочтительное имя — это имя, используемое международными стандартами (если они есть). Если предпочтительное имя из международного стандарта превышает максимальную длину, допустимую для данного атрибута, то оно должно быть сокращено без потери смысла.

Oblig: Обязательный.

Trans: Необходим перевод.

For: 255, буквенно-цифровой.

Mapp: Отображение **property_DET\class_and_property_element.names\item_name.preferred_name**

Expl: Диаметр резьбы

7.2.5 Сокращенное имя

Obj: Дать имя свойству для его представления в ограниченном пространстве.

Descr: Это есть смысловое сокращение предпочтительного имени. Если существуют стандартные аббревиатуры, то они должны использоваться. Сокращенное имя может совпадать с предпочтительным именем или быть буквенным обозначением.

Oblig: По выбору.

Trans: Необходим перевод в случае, когда сокращенное имя идентично буквенному обозначению. Оно должно быть одним и тем же на всех языках.

For: 30, буквенно-цифровой.

Mapp: Отображение **property_DET\class_and_property_elements.names\item_name.short_name**

Expl: Пример сокращенного имени: **thread_diam** (диаметр резьбы).

7.2.6 Предпочтительное буквенное обозначение

Obj: Укоротить имя свойства. Если оно существует, то используется вместо сокращенного имени в таблицах, формулах, на чертежах и т. д.

Descr: Буквенное обозначение является уникальным внутри характеристического класса продукции. Его берут из международных стандартов. Предпочтительное буквенное обозначение всегда нужно в текстовом представлении. Его также используют в текстовых математических представлениях MathML.

Пример — Источниками буквенных обозначений могут быть ИСО 80000/МЭК 80000 (ранее ИСО 31), МЭК 60027, МЭК 60748 и стандарты на продукцию.

Oblig: По выбору.

Trans: Перевода не требуется.

For: Буквенно-цифровой и формат MathML (если есть).

Mapp: Отображение **property_DET.preferred_symbol**

7.2.7 Синонимические буквенные обозначения

Obj: Сократить имя свойства. Используется в таблицах, формулах, на чертежах и т. д.

Descr: Буквенное обозначение является уникальным внутри характеристического класса продукции. Берется из Международных стандартов.

Пример — Примеры таких стандартов: ИСО 80000/МЭК 80000 (ранее ИСО 31), МЭК 60027, МЭК 60748, стандарты на продукцию.

Допускается отсутствие, одно или несколько синонимических буквенных обозначений. Синонимическое буквенное обозначение всегда используется в текстовом представлении. Оно используется в представлении MathML.

Oblig: По выбору.

Trans: Перевода не требуется.

For: Буквенно-цифровой формат, формат MathML (если есть).

Mapp: отображение **property_DET.synonymous_symbol**

7.2.8 Синонимическое имя

Obj: Дать синонимы предпочтительным именам на основании имен, используемых по локальным или историческим причинам.

Descr: Альтернативное обозначение, отличающееся от данного предпочтительного имени, но представляющее то же понятие. Синонимических имен может не быть, может быть одно или несколько.

Oblig: По выбору.

Trans: Необходим перевод.

For: 255, буквенно-цифровой.

Mapp: Отображение **property_DET\class_and_property_elements.names\item_names.synonymus_names**

7.2.9 Классификация типов свойств

Obj: Классифицировать различные свойства для облегчения пользования большими наборами свойств.

Descr: Все множество свойств разделено на подмножества в соответствии с категориями, определенными в ИСО 80000/МЭК 80000 (ранее ИСО 31). Атрибутом классификации типов свойств является ссылка на категорию ИСО 80000/МЭК 80000, соответствующую данному свойству.

Oblig: По выбору.

Trans: Перевода не требуется.

For: Одна заглавная буква и две цифры.

Mapp: Отображение **property_DET.DET_classification**

Примечание — Обзор основных классов и категорий свойств ИСО 80000/МЭК 80000 приведен в приложении А. Классификация количественных мер ИСО 80000/МЭК 80000 обсуждается в приложении Н. Классификация неколичественных свойств ИСО 80000/МЭК 80000 (идентификации и индикаторы) приведена в приложении А.

7.2.10 Определение

Obj: Описать смысл свойства.

Descr: Определение, описывающее смысл свойства и позволяющее отличить его от всех других свойств. Это описание должно быть полным и неоднозначным. Существенные слова не должны иметь омонимов и синонимов.

Oblig: Обязательный.

Trans: Необходим перевод.

For: Неограниченная буквенно-цифровая строка.

Mapp: Отображение **property_DET\class_and_property_element.definition**

7.2.11 Исходный документ для определения

Obj: Дать ссылку на исходный документ, из которого было получено определение свойства.

Descr: Как минимум, ссылка должна содержать № документа и дату выпуска документа.

Oblig: По выбору.

Trans: Необходим перевод.

Примечание — Если документ представлен как идентифицированный документ identified_document, то имя документа может быть приведено на разных языках. Если обмен соответствует ИСО 13584-25 или ИСО 13584-35 (OntoML), то документами можно обмениваться на разных языках.

For: Буквенно-цифровой. Идентификатор документа.

Mapp: Отображение **property_DET\class_and_property_element.source_doc_of_definition\identified_document.document_identifier**

7.2.12 Примечание

Obj: Дать дополнительную информацию по любой части определения, которая является существенной для понимания этого определения.

Descr: Копируется из исходного определения в документе в определение свойства.

Oblig: По выбору.

Trans: Необходим перевод.

For: Неограниченная буквенно-цифровая строка.

Mapp: Отображение **property_DET\class_and_property_elements.note**

7.2.13 Замечание

Obj: Пояснительный текст для углубления понимания смысла использования свойства.

Descr: Свободное текстовое замечание. Оно не должно искажать смысл свойства.

Oblig: По выбору.

Trans: Необходим перевод.

For: Неограниченная буквенно-цифровая строка.

Mapp: Отображение **property_DET\class_and_property_elements.remark**

7.2.14 Единица измерения

Obj: Назначить по умолчанию единицу измерения и, возможно, альтернативные единицы измерения, в которых выражаются значения количественных свойств.

Descr: Формальная модель одной или нескольких единиц измерения и/или одной или нескольких ссылок на словарь единиц измерения, откуда можно взять формальную модель единицы измерения.

Примечание 1 — Идентификаторы, используемые для ссылок на словарь единиц измерения, позволяют загрузить формальную модель из вспомогательной базы, соответствующей ИСО/ТС 29002-20.

Oblig: Обязательный (для количественных данных).

Trans: Перевода не требуется.

For: Единицы измерения представляются в соответствии с ИСО 10303-41. При необходимости используются расширения, установленные в общей словарной модели ИСО 13584/МЭК 61360. Также могут использоваться математические строки. Математические строки всегда имеются в текстовых представлениях. Они могут быть в представлениях MathML. Ссылка на словарь единиц измерения выполняется с помощью идентификаторов, соответствующих серии ИСО 29002.

Mapp: Отображения **property_DET.domain\int_measure_type.unit** или **property_DET.domain\real_measure_type.unit** или **property_DET.domain\rational_measure_type.unit** и/или **property_DET.domain\int_measure_type.unit_id** или **property_DET.domain\real_measure_type.unit_id** или **property_DET.domain\rational_measure_type.unit_id** и/или **property_DET.domain\int_measure_type.alternative_units** или **property_DET.domain\real_measure_type.alternative_units** или **property_DET.domain\rational_measure_type.alternative_units** и/или **property_DET.domain\int_measure_type.alternative_unit_ids** или **property_DET.domain\real_measure_type.alternative_unit_ids** или **property_DET.domain\rational_measure_type.alternative_unit_ids**

Примечание 2 — Рассматриваемая модель данных допускает альтернативные единицы измерения. Но если альтернативные единицы измерения разрешены, то они прописываются в определении свойства, и каждое значение свойства, соответствующее какой-то альтернативной единице измерения, ассоциируется с этой единицей.

7.2.15 Условие

Obj: Формально идентифицировать контекстный параметр, от которого зависят контекстно-зависимые характеристики.

Descr: Описание кодов адекватных контекстных параметров.

Oblig: Обязательно для контекстно-зависимых характеристик.

Trans: Перевода не требуется.

For: Ссылка на набор базовых семантических блоков.

Mapp: Отображение **property_DET\dependent_P_DET.depends_on**

7.2.16 Формула

Obj: Правило или утверждение в математической форме, выражающее семантику количественного свойства. Формула не должна изменять какую-либо существенную информацию о смысле рассматриваемого определения.

Descr: Математическое выражение определения свойства.

Oblig: По выбору.

Trans: Перевода не требуется.

For: Математическая строка (математические строки всегда присутствуют в текстовом представлении; они также бывают в текстовом представлении MathML).

Mapp: Отображение **property_DET.formula**

7.2.17 Формат значения

Obj: Спецификация длины и шаблон рекомендуемого представления значения свойства с типами данных: строка **string_type** (либо ее подтип), число **number_type** (либо его подтип), набор значений (определенных либо типом **level_type**, либо укрупненными типами данных: **list_type**, **set_type**, **bag_type**, **array_type** или **set_with_subset_constraint_type**). При необходимости, данный атрибут дает рекомендацию системе о порядке представления строчных и числовых значений. Данная рекомендация не должна противоречить типу данных и возможным ограничениям на значения свойств. Значение данного атрибута должно быть совместимым с типом данных свойств: оно не должно изменять указанный тип данных, в противном случае оно игнорируется. Значение данного атрибута также должно быть совместимым с допустимыми ограничениями свойств, определенными на значениях свойств с помощью схемы ограничений, установленной в настоящем стандарте. В противном случае значение данного атрибута игнорируется.

Примечание 1 — Типы данных list_type, set_type, bag_type, array_type и set_with_subset_constraint_type определены в ИСО 13584-25.

Descr: При необходимости формат значения определяется в соответствии с одним из определений высоты (см. ниже). Синтаксис указанных форматов определен в приложении D с помощью подмножества расширенной формы Бекуса-Наура (EBNF), установленного в ИСО/МЭК 14977.

a) Форматы значений неколичественных типов данных: 5 определений. Они предназначены для свойств с неколичественными типами данных, например, строка **string_type** (или один из ее подтипов) или наборы таких значений.

Алфавитный формат значения (A),

Смешанный символьный формат значений (M),

Числовой формат значений (N),

Смешанный алфавитный и числовой символьный формат значений (X),

Бинарный формат значений (B).

b) Форматы значений количественных типов данных: 3 определения. Они предназначены для свойств с числовыми значениями, т.е. числовой тип **number_type** (или один из его подтипов) или наборы таких значений.

Целочисленный формат значений (NR1),

Действительный десятичный формат значений (NR2),

Действительный десятичный и показательный формат значений (NR3),

Рациональный формат значений (NR4).

Oblig: По выбору.

Trans: Перевода не требуется.

For: 80, буквенно-цифровой.

Mapp: Отображение **property_DET.domain\simple_type.value_format**

Примечание 2 — Для поименованных типов данных формат значения относится к значениям со ссылочным атрибутом **referred_type** поименованного типа **named_type**.

7.2.18 Дата исходного определения

Obj: Показать, когда свойство было определено поставщиком информации, т.е. когда оно было объявлено действительным данным поставщиком. Данная дата никогда не изменяется. Ее можно использовать для проверки.

Descr: В соответствии с ИСО 8601.

Oblig: По выбору.

Trans: Перевода не требуется.

For: 10, буквенно-цифровой.

Mapp: Отображение **property_DET\dictionary_element.time_stamps\dates.date_of_original_definition**

Expl: Пример даты: 1967-08-20.

7.2.19 Дата текущей версии

Obj: Показать дату определения текущей версии.

Descr: В соответствии с ИСО 8601.

Oblig: По выбору.

Trans: Перевода не требуется.

For: 10, буквенно-цифровой.

Mapp: Отображение **property_DET\dictionary_element.time_stamps\dates.date_of_current_version**

7.2.20 Дата текущего пересмотра

Obj: Показать дату изменения последнего № пересмотра.

Descr: В соответствии с ИСО 8601.

Oblig: По выбору

Trans: Перевода не требуется.

For: 10, буквенно-цифровой.

Mapp: Отображение **property_DET\dictionary_element.time_stamps\dates.date_of_current_revision**

7.2.21 № версии

Obj: Описать каждую версию свойства. Новый № версии свойства создается при изменении атрибутов, определяющих порядок использования данного свойства.

Примечание 1 — Изменения не допускаются, если они искажают смысл свойства.

Примечание 2 — Изменения свойств, приводящие к изменению № версии, определены в разделе 9.

Descr: Строка, содержащая натуральное число, указывающее на различные версии свойства в течение жизненного цикла. Номера версий указываются в возрастающем порядке. Новая версия свойства генерируется в соответствии с правилами раздела 9.

Oblig: Обязательный.

For: 10, численный.

Trans: Перевода не требуется.

Mapp: Отображение **property_DET\dictionary_element.identified_by\basic_semantic_unit.version**

7.2.22 № пересмотра

Obj: Символизировать каждый пересмотр заданной версии свойств. Новый номер пересмотра свойства создается, если изменение некоторого атрибута (описывающего данное свойство) не искажает смысл свойства и не изменяет порядок его использования.

Примечание 1 — Изменение недопустимо, если оно искажает смысл свойства.

Примечание 2 — Изменения свойства, приводящие к изменению номера пересмотра, определены в разделе 9.

Descr: Строка, содержащая натуральное число, используемая для административного контроля свойства. Последовательные номера пересмотров должны идти в возрастающем порядке для каждого нового значения версии свойства. Для каждого свойства (с уникальным идентификатором) может быть действительным только один текущий номер пересмотра в любой момент времени. Новый номер пересмотра свойства генерируется в соответствии с определением раздела 9. Если выпускается новая версия, то номер пересмотра ставится на ‘0’.

Oblig: Обязательный.

For: 3, численный.

Trans: Перевода не требуется.

Mapp: Отображение **property_DET\dictionary_element.revision**

7.2.23 Не рекомендуется для использования

Obj: Указать, можно ли свойство использовать по-прежнему, или его нельзя использовать с новыми характеристиками.

Примечание 1 — Не рекомендуемые для использования свойства можно изучать по ссылочному словарю, дающему интерпретацию характеристик, определенных в предшествующих версиях данного словаря.

Descr: Булево значение (по выбору). Если оно равно **true**, то рассматриваемое свойство, возможно, использовалось в прошлом, но оно не должно использоваться далее с новыми характеристиками.

Примечание 2 — Если данный атрибут не существует, то свойство не рекомендуется для использования.

Oblig: По выбору.

For: Булево значение.

Trans: Перевода не требуется.

Mapp: Отображение **property_DET\dictionary_element.is_deprecated**

7.2.24 Интерпретация, не рекомендуемая для использования

Obj: Показать нецелесообразность использования, а также порядок интерпретации значений, не рекомендуемых для использования свойств.

Descr: Пояснение, которое является обязательным, когда свойство является не рекомендуемым для использования.

Oblig: По выбору.

For: Переводимый текст.

Trans: Необходим перевод.

Mapp: Отображение **property_DET\dictionary_element.is_deprecated_interpretation**

7.2.25 Административные данные

Obj: Запись информации о жизненном цикле свойства.

Descr: Сущность, представляющая исходный язык (на котором было приведено описание свойства), список языков (на которые переведено описание данного свойства) и допустимый статус, определяющий состояние жизненного цикла свойства.

Oblig: По выбору.

For: Сущность **administrative_data**.

Trans: Перевода не требуется.

Mapp: Отображение **property_DET\dictionary_element.administration**

8 Словарные элементы, описывающие классы продукции

8.1 Отображение классов на общую словарную модель ИСО 13584/МЭК 61360

В общей словарной модели ИСО 13584/МЭК 61360 характеристический класс отображается на класс **class_BSU** (для идентификации) и на словарный элемент класса **item_class** (для описания). Обе реализации продукции и аспекты продукции отображаются на класс **item_class**.

Пример 1 — Головка винта является особенностью. В ИСО 13584-511 она описана с помощью класса элементов item_class head (головка).

Сущность **item_class** включает атрибут **instance_sharable,** описывающий концептуальный статус элемента: это либо отдельный элемент (**instance_sharable = *true***), либо общий (**instance_sharable = *false***). Значение данного атрибута не подразумевает наличие каких-либо ограничений на уровне представления данных.

Пример 2 — На один и тот же экземпляр класса головок винтов можно ссылаться из нескольких экземпляров класса винтов. Это означает, что существует несколько головок винтов, но все эти головки винтов имеют один и тот же характеристический класс и один и тот же набор значений свойств.

Примечание 1 — Два подтипа класса элементов **item_class** (класс компонентов **component_class** и класс материалов **material_class**) определены внутри словарной модели первого издания ИСО 13584-42 и МЭК 61360-2. Указанные подтипы не рекомендуются для использования, они удалены из настоящего стандарта.

Примечание 2 — Нижеследующие изменения гарантируют, что словарь, соответствующий первому изданию настоящего стандарта, удовлетворяет требованиям данного издания: (1) замена **component_class** и **material_class** на **item_class** с помощью ссылочного словаря; (2) для каждого нового класса **item_class** добавляется атрибут **instance_sharable** со значением ***true***.

Примечание 3 — Другой подтип класса элементов **item_class (feature_class)** введен в ИСО 13584-24:2003. Данный подтип также не рекомендуется для использования в новых практических реализациях ИСО 13584-42 и МЭК 61360-2.

Примечание 4 — Нижеследующие изменения гарантируют, что словарь, соответствующий первому изданию настоящего стандарта, удовлетворяет требованиям данного издания: (1) замена **feature_class** на **item_class** с помощью ссылочного словаря; (2) для каждого нового класса **item_class** добавляется атрибут **instance_sharable** со значением *false*.

В общей словарной модели ИСО 13584/МЭК 61360 класс категоризации отображается на **class_BSU** (для идентификации) и на словарный элемент класса категоризации **categorization_class** (для описания). Для класса категоризации используется только поднабор атрибутов характеристического класса. Заданный атрибут класса категоризации, называемый суперклассом **categorization_superclasses**, позволяет описывать суперклассы в условной иерархии соотношений включения в класс.

Примечание 5 — Условные соотношения для характеристических классов позволяют также импортировать свойства в условные классы. Поэтому данные соотношения представляются с помощью различных атрибутов. См. определение, приведенное в приложении F настоящего стандарта.

В данном разделе описаны основные атрибуты, связанные с классом прямо или косвенно.

8.2 Атрибуты

Каждый характеристический класс продукции описывается словарным элементом, содержащим нижеследующие атрибуты:

- код;
- суперкласс;
- предпочтительное имя;
- сокращенное имя;
- синонимическое имя;
- видимые типы;
- применимые типы;
- свойства со значением, определенным классом;
- видимые свойства;
- применимые свойства;
- определение значения для класса;
- определение;
- исходный документ для определения;
- примечание;
- замечание;
- упрощенный чертеж;
- дата исходного определения;
- дата текущей версии;
- дата текущего пересмотра;
- номер версии;
- номер пересмотра;
- ограничения;
- не рекомендуется для использования;
- интерпретация нецелесообразности использования;
- административные данные.

Более того, словарный элемент, описывающий класс категоризации продукции, также содержит следующий атрибут:

- **categorization_class_superclass**.

Запись каждого атрибута структурируется следующим способом (записи может не быть, если сущность неприменима):

- Имя атрибута;

- Obj: Цель;
- Descr: Описание;
- Oblig: Обязательство;
- Trans: Необходимость перевода на другой язык;
- For: Формат представления, максимальное число (тип) символов (для строк);
- Mapp: Отображение на конструктив ресурса, используемое в общей словарной модели ИСО 13584/МЭК 61360;
- Expl: Пример.

8.2.1 Код

Obj: Идентификация характеристических классов (классов категоризации) продукции и обеспечение абсолютной идентификации внутри словаря данных, ассоциированных с кодом поставщика информации и с номером версии.

Descr: Базовая семантическая единица, ассоциированная с характеристическим классом продукции.

Oblig: Обязательный.

Trans: Перевода не требуется.

Mapp: Отображение **item_class\dictionary_element.identified_by\basic_semantic_unit.code**

8.2.2 Суперкласс

Obj: Ссылка на ближайший уникальный родительский характеристический класс с помощью представительного соотношения (is-a relationship) характеристического класса.

Descr: Базовая семантическая единица ближайшего родительского характеристического класса для текущего характеристического класса.

Oblig: По выбору (если не существует, то класс не имеет суперкласса).

Trans: Перевода не требуется.

For: Базовая семантическая единица для класса.

Mapp: **item_class\class.its_superclasses**

Примечание 1 — При определении иерархии стандартных характеристик один или несколько узлов могут не иметь суперкласса. Данные узлы являются корнем (корнями) иерархии.

Примечание 2 — Рекомендуется, чтобы корни иерархий стандартных характеристик (а также некоторые другие узлы) ссылались на некоторые узлы Международной Классификации ИСО/МЭК для предварительно определенного дерева стандартов (ICS) с помощью *условного* соотношения для уяснения целевой области, адресуемой каждой иерархией (подиерархией) характеристик. На другие иерархии классификации можно, при необходимости, ссылаться аналогично.

Примечание 3 — При ссылке на класс ICS используемый код класса — это код, определенный для данного класса в указанном ICS, при этом точки заменяются подчеркиванием. Номер версии используемого класса равен '001'. В соответствии с ИСО 13584-26 код поставщика, используемый в 1-м издании ICS от 1992 г., равен «112/1///_00_1». Код поставщика, используемый в 6-м издании от 2005 г., равен «112/1///_00_6». Поставщикам словарных данных рекомендуется использовать более поздние издания ICS. Другие издания должны идентифицироваться кодом поставщика типа «112/1///_00_<№ издания ICS>», где <№ издания ICS> — это целое число, т. е. номер ссылочного ICS.

8.2.3 Предпочтительное имя

Obj: Дать существенное (возможно более подробное) описание характеристического класса (класса категоризации) продукции. Используется для связи и более глубокого понимания.

Descr: Предпочтительное имя идентично имени, используемом в международных стандартах (если они есть). Если длина предпочтительного имени в международном стандарте превышает максимально допустимую длину, то данный атрибут должен быть осмысленно сокращен.

Oblig: Обязательный.

Trans: Необходим перевод.

For: 255, буквенно-цифровой.

Mapp: **class\class_and_property_element.names\item_name.preferred_name**

Expl: Резьбы винтов.

8.2.4 Сокращенное имя

Obj: Дать имя характеристическому классу (классу категоризации) продукции в ограниченном пространстве.

Descr: Это есть осмысленное сокращение предпочтительного имени. Если стандартные аббревиатуры уже существуют, то они должны быть использованы. Они могут быть идентичны предпочтительному имени.

Oblig: По выбору.

Trans: Необходим перевод. Если сокращенное имя идентично буквенному обозначению, то оно должно быть одним и тем же на всех языках.

For: 30, буквенно-цифровой.

Mapp: Отображение **class\class_and_property_elements.names\item_names.short.name**

8.2.5 Синонимические имена

Obj: Синонимы предпочтительного имени характеристического класса (класса категоризации) способствуют переходу от имен, используемых по локальным или историческим причинам.

Descr: Альтернативное обозначение, которое отклоняет имеющееся предпочтительное имя, сохраняет его смысл. Синонимов может не быть, а также может быть один или несколько.

Oblig: По выбору.

Trans: Необходим перевод.

For: 255, буквенно-цифровой.

Mapp: Отображение **class\class_and_property_elements.names\item_names.synonymus_names**

8.2.6 Видимые типы данных

Obj: Определить новые поименованные типы, на которые можно ссылаться как на типы данных, с помощью видимых свойств данного характеристического класса (какого-либо подкласса) (видимый тип).

Descr: Различные свойства могут иметь один и тот же тип данных.

Пример — Набор кодов, идентифицирующих материалы.

Указанные типы данных могут быть определены как поименованные типы, независимо от рассматриваемого свойства. На них можно потом сослаться как на типы данных различных свойств. Определение поименованного типа должно содержать код, номер версии и тип данных поименованного типа, установленного путем использования конструктивов ресурса общей словарной модели ИСО 13584/МЭК 61360. Для неколичественного типа кода **non_quantitative_code_type** и неколичественного целого типа **non_quantitative_int_type** тип данных должен устанавливаться как набор словарных значений **dic_values,** составляющих уникальный код и набор имен (переведенных по возможности).

Oblig: По выбору.

Trans: Для типов **non_quantitative_code_type** и **non_quantitative_int_type** имена значений требуют перевода.

For: Форматы **data_type_BSUs** и **data_type_elements**

Mapp: Отображение **item_class\dictionary_element.identified_by\class_BSU.added_visible_data_types**

Примечание 1 — Видимые типы наследуются.

Примечание 2 — На видимые типы, не являющиеся применимыми типами, можно ссылаться для определения типа данных видимых свойств характеристического класса (какого-либо из его подклассов).

Пример — «Материал» поименованного типа named_type может быть определен на уровне некоторого характеристического класса как тип неколичественного кода non_quantitative_code_type. Набор значений данного поименованного типа состоит из набора кодов, ассоциированных с различными (переведенными) именами. На данный поименованный тип можно ссылаться в некоторых характеристических классах (подклассах) для определения типов данных свойства, соответствующего материалу, из которого состоит продукт рассматриваемого характеристического класса. На него можно ссылаться также в некоторых других характеристических классах (подклассах) для определения типов данных свойства, соответствующего покрытию продукции рассматриваемого характеристического класса.

8.2.7 Применимые типы

Obj: Определить видимые типы, допустимые как типы данных для свойств, применимых для характеристического класса (или какого-либо его подкласса).

Descr: Список кодов определенных (унаследованных) видимых типов характеристического класса, которые становятся применимыми типами для рассматриваемого характеристического класса (и какого-либо его подкласса).

Oblig: По выбору.

Trans: Перевода не требуется.

For: Список **data_type_BSU**.

Mapp: Отображение **item_class\class.defined_types**

8.2.8 Свойства со значением, определенным классом

Obj: Определить, какие свойства могут быть назначены одному значению в характеристическом классе продукции, где данное свойство объявлено как свойство со значением, определенным классом (каким-либо его подклассом).

Descr: Свойства со значением, определенным классом, способствуют представлению: (1) различных перспектив для набора продукции, принадлежащего рассматриваемому характеристическому классу, (2) свойств, принимающих одно значение для всех видов продукции, принадлежащей заданному классу. Если свойство со значением, определенным классом, определено на уровне некоторого характеристического класса, то пользователь словаря может запросить все подклассы данного класса, где этому свойству присвоено некоторое значение.

Oblig: По выбору.

Trans: Перевода не требуется.

For: Список **property_BSU.**

Mapp: Отображение **item_class.sub_class_properties**

Примечание — Свойства со значением, определенным классом, являются унаследованными. Они не появляются в списке свойств **sub_class_properties** подклассов.

8.2.9 Видимые свойства

Obj: Описать новые свойства, определенные на уровне рассматриваемого характеристического класса. Они, таким образом, могут быть описаны как применимые для данного характеристического класса или какого-либо его подкласса (видимые свойства).

Descr: Новые свойства (с учетом наследственности) продукции, принадлежащие к нелистовому или листовому характеристическому классу, могут обладать или не обладать особенностями подклассов, которым они принадлежат.

For: Набор свойств **property_BSU.**

Oblig: Обязательный (возможно, пустой).

Mapp: Отображение **item_class\dictionary_element.identified_by\class_BSU.added_visible_properties**

8.2.10 Применимые свойства

Obj: Описать новые свойства, определенные как применимые для рассматриваемого характеристического класса (какого-либо его подкласса).

Descr: Новые свойства (с учетом наследственности), которыми должна обладать продукция, принадлежащие нелистовому или листовому характеристическому классу. Свойства данного списка должны быть видимыми для данного характеристического класса, т. е. они должны быть определены как видимые либо данным характеристическим классом, либо характеристическим классом более высокого уровня в рассматриваемой иерархии. Порядок перечисления в списке должен по умолчанию соответствовать порядку представления свойств в любом случае, когда такой порядок определен.

Пример — Представление свойств некоторого класса на экране.

For: Список свойств **property_BSU.**

Oblig: Обязательный (возможно, пустой).

Mapp: Отображение **item_class\class.described_by**

8.2.11 Значения, постоянные в классе

Obj: Определить значения, назначенные некоторым свойствам со значением, определенным классом, объявленным в рассматриваемом характеристическом классе или в каком-либо характеристическом классе на более высоком иерархическом уровне.

Descr: Свойства со значением, определенным классом, предназначены для удовлетворения запросов пользователей на выбор такого набора характеристических классов, для которых указанное свойство имеет заданное значение. Если значение задано для свойства со значением, определенным классом, то данное значение является унаследованным и не должно переопределяться.

For: Набор заданных значений **class_value_assignments**.

Oblig: По выбору.

Mapp: Отображение **item_class.class_constant_values**

8.2.12 Определение

Obj: Пояснить смысл характеристического класса (класса категоризации), путем указания его назначения в письменной форме.

Descr: Утверждение, описывающее смысл характеристического класса, позволяет установить его отличие от всех других характеристических классов. Данное утверждение должно быть определением в том смысле, что оно должно быть полным и однозначным. Все существенные слова не должны иметь омонимов и синонимов.

Oblig: Обязательный.

Trans: Необходим перевод.

For: Формат **class\class_and_property_element.definition**

8.2.13 Исходный документ для определения

Obj: Дать ссылку на исходный документ, из которого получено определение характеристического класса (класса категоризации).

Descr: Как минимум, ссылка должна содержать номер документа и дату его выпуска.

Oblig: По выбору.

Trans: Перевода не требуется.

For: Буквенно-цифровой. Идентификатор документа.

Mapp: Отображение **class\class_and_property_elements.**
source_doc_of_definition\identified_document.document_identifier

Примечание — Если обмен соответствует ИСО 13584-24:2003, то документом можно обмениваться.

8.2.14 Примечание

Obj: Дополнительная информация по какой-либо детали определения, существенной для его понимания.

Descr: Текстовое утверждение в произвольной форме.

Oblig: По выбору.

Trans: Необходим перевод.

For: Неограниченная буквенно-цифровая строка.

Mapp: Отображение **item_class\class_and_property_element.note**

8.2.15 Замечание

Obj: Пояснительный текст, приводящий дополнительную информацию о смысле использования характеристического класса (класса категоризации) продукции.

Descr: Свободные текстовые замечания не должны искажать смысл основного текста.

Oblig: По выбору.

Trans: Необходим перевод.

For: Неограниченная буквенно-цифровая строка.

Mapp: Отображение **item_class\class_and_property_elements.remark**

8.2.16 Упрощенный чертеж

Obj: Получить визуализацию (по запросу пользователя), предоставляющую образ характеристического класса (класса категоризации) продукции.

Descr: Чертеж, включающий, по крайней мере, ссылочную систему координат рассматриваемой продукции (используемую для всех представлений данного вида продукции) и буквенные обозначения основных применимых свойств.

Oblig: По выбору.

Trans: Перевода не требуется.

For: Графическая сущность.

Mapp: Отображение **item_class.simplified_drawing**

Примечание — Графическую сущность можно обменивать как сущность внешней графики **external_graphics,** описанную в настоящем стандарте. Указанную сущность **external_graphics** можно представить как гипертекстовый файл **http_file,** определенный в ИСО 13584-24:2003, на который ссылается словарная схема **ISO13584_IEC61360_dictionary_schema,** установленная в приложении F.

8.2.17 Дата исходного определения

Obj: Показать, когда характеристический класс (класс категоризации) продукции был определен поставщиком информации и когда он был объявлен действительным указанным поставщиком. Указанная дата никогда не изменяется. Она может быть использована для сверки.

Descr: В соответствии с ИСО 8601.

Oblig: По выбору.

Trans: Перевода не требуется.

For: 10, буквенно-цифровой.

Mapp: Отображение **class\dictionary_element.time_stamps.date_of_original_definition**
Expl: 1967-08-20
8.2.18 Дата текущей версии
Obj: Показать дату определения текущей версии.
Descr: В соответствии с ИСО 8601.
Oblig: По выбору.
Trans: Перевода не требуется.
For: 10, буквенно-цифровой.
Mapp: Отображение **class\dictionary_element.time_stamps.date_of_current_version**
8.2.19 Дата текущего пересмотра
Obj: Показать дату изменения последнего номера пересмотра.
Descr: В соответствии с ИСО 8601.
Oblig: По выбору.
Trans: Перевода не требуется.
For: 10, буквенно-цифровой.
Mapp: Отображение **class\dictionary_element.time_stamps.date_of_current_revision**
8.2.20 № версии
Obj: Охарактеризовать каждую версию характеристического класса (класса категоризации). Новый номер версии класса определяется, если изменение атрибутов, описывающих данный класс, влияет на порядок его использования.

Примечание 1 — Изменение недопустимо, если оно искажает смысл класса.

Примечание 2 — Изменения класса, приводящие к изменению номера версии, определены в разделе 9.

Descr: Строка, содержащая натуральное число, указывающее на различные версии характеристического класса в течение его жизненного цикла. Номера версий генерируются в возрастающем порядке. Новый номер версии характеристического класса генерируется в соответствии с определением, приведенным в разделе 9.
Oblig: Обязательный.
For: 10, числовой.
Trans: Перевода не требуется.
Mapp: Отображение **item_class\dictionary_element.identified_by\basic_semantic_unit.version**
8.2.21 Номер пересмотра
Obj: Охарактеризовать каждую версию характеристического класса (класса категоризации). Новый номер пересмотра класса определяется, когда изменения атрибутов, описывающих данный класс, не искажают смысл класса и не нарушают порядок его использования.

Примечание 1 — Изменение недопустимо, если оно искажает смысл класса.

Примечание 2 — Изменения в классе, оказывающие влияние на номер пересмотра, определены в разделе 9.

Descr: Строка, содержащая натуральное число, используемое для административного контроля класса. Последовательные номера пересмотров идут в возрастающем порядке для каждого значения номера версии класса. Для каждого класса (с уникальным идентификатором) только один номер пересмотра может быть текущим в любой момент времени. Новый номер пересмотра класса генерируется в соответствии с определением, данным в разделе 9. Если вышла новая версия, то номер пересмотра устанавливается на '0'.
Oblig: Обязательный.
For: 3, числовой.
Trans: Перевода не требуется.
Mapp: Отображение **item_class\dictionary_element.revision**
8.2.22 Ограничения
Obj: Ограничить целевые области значений некоторых свойств характеристического класса до некоторых подмножеств их унаследованных областей значений.

Примечание 1 — Данные ограничения применяются в классе, для которого они определены, а также во всех его представительных подклассах и условных классах.

Примечание 2 — Каждое ограничение может быть ассоциировано с идентификатором, соответствующим ИСО/ТС 29002-5. Данный идентификатор включает идентификатор поставщика, называемый RI, идентификатор данных, называемый DI, и идентификатор версии, называемый VI.

Descr: Набор ограничений.

Oblig: По выбору.

For: Набор ограничений.

Trans: Перевода не требуется.

Mapp: Отображение **item_class\class.constraints**

8.2.23 Реализация коллективного доступа

Obj: Описать, является ли класс элементов отдельным элементом или это зависимый элемент, который может быть только фактическим компонентом другого элемента.

Примечание — Данный атрибут не требует какой-либо особой практической реализации на уровне представления данных.

Descr: Булев атрибут, определенный для класса элементов **item_class.**

Oblig: По выбору.

For: Булев.

Trans: Перевода не требуется.

Mapp: Отображение **item_class.instance_sharable**

8.2.24 Суперклассы класса категоризации

Obj: Определить, какие из классов категоризации стоят одной ступенью выше в иерархии условных соотношений включения в класс.

Примечание — Данный атрибут применяется только для классов категоризации.

Descr: Базовые семантические единицы класса для ближайших родительских классов категоризации в иерархии условных соотношений включения в класс.

Oblig: По выбору.

For: Набор классов **class_BSU**.

Trans: Перевода не требуется.

Mapp: Отображение **categorization_class.categorization_class_superclasses**

8.2.25 Не рекомендуемый для использования

Obj: Указать, можно ли класс по-прежнему использовать или его больше не следует использовать в новых условиях.

Примечание 1 — Не рекомендуемые для использования классы могут обслуживаться в ссылочных словарях, чтобы дать возможность пользователю интерпретировать характеристики, определенные в предшествующих версиях данного словаря.

Descr: Булево значение по выбору: если оно равно **true**, то указанный класс, возможно, использовался в прошлом, но не должен далее использоваться в новых условиях.

Примечание 2 — Если данный атрибут не существует, то класс не рекомендуется для использования.

Oblig: По выбору.

For: Булево значение.

Trans: Перевода не требуется.

Mapp: Отображение **class\dictionary_element.is_deprecated**

8.2.26 Интерпретация нецелесообразности использования

Obj: Указать нецелесообразность использования некоторого класса и определить порядок интерпретации значений рассматриваемого не рекомендуемого для использования класса.

Descr: Обязательное примечание в случае, если класс не рекомендуется для использования.

Oblig: По выбору.

For: Переводимый текст.

Trans: Необходим перевод.

Mapp: Отображение **class\dictionary_element.is_deprecated_interpretation**

8.2.27 Административные данные

Obj: Зарегистрировать информацию о жизненном цикле класса.

Descr: Сущность, представляющая исходный язык (на котором приведено определение класса), список языков (на которые переводится описание класса), допустимый статус, определяющий состояние жизненного цикла класса.

Oblig: По выбору.

For: Сущность «административные данные» **administrative_data**.

Trans: Перевода не требуется.

Mapp: Отображение **class\dictionary_element.administration**

9 Правила управления изменением словаря

Данный раздел определяет правила организации, контроля и отслеживания изменений ссылочных словарей.

Пусть имеется одна версия O_t ссылочного словаря и набор описаний продукции, основанных на другой версии $O_{t'}$ данного словаря. Цели рассматриваемых правил: (1) установить порядок принятия решения о том, что имеющаяся версия словаря O_t позволяет корректно интерпретировать описания продукции, основанные на другой версии O_t' словаря; в противном случае (2) установить порядок выборки части словаря O_t, которую следует обновить, чтобы обеспечить корректную интерпретацию имеющихся описаний продукции.

Примечание — Все обсуждение правил управления словарем, представленных в настоящем разделе, основано на характеристиках продукции. В настоящем обсуждении не рассматриваются классы категоризации, не используемые для характеризации продукции. Таким образом, в данном разделе понятие «класс» означает «характеристический класс», а понятие «словарный элемент» означает «все словарные элементы, кроме классов категоризации». Особое правило для классов категоризации — это правило 8.

9.1 Принцип онтологической непрерывности

Цель онтологии в серии ИСО 13584, называемой ссылочным словарем для данной области состоит в том, чтобы установить:

- порядок обмена неоднозначной информацией о продукции между бизнес-партнерами;
- порядок хранения устойчивых характеристик продукции в нескольких непрерывно обновляемых архивах данных.

Используемый метод — кодирование каждого вида продукции. Характеристика продукции включает:

- характеристический класс, которому принадлежит продукт;
- набор пар значений свойств, откуда требуемые свойства выбираются из свойств, применимых к данному характеристическому классу.

Пример — Пусть продукт представлен с помощью ссылочного словаря крепежных элементов по ИСО 13584-511, например, это колпачковая гайка с номинальным диаметром 5 и высотой 4, т.е.:

cap nut (nominal diameter = 5; heigh of nut = 4)

Данная характеристика представляет продукт, который является «шестигранной гайкой, заглушенной с одной стороны плоским колпачком». «Номинальный диаметр резьбы» равен 5 мм, «габаритная высота гайки» равна 4 мм.

Примечание 1 — В электронном формате обмена как характеристический класс, так и свойства, появляющиеся при описании, представляются с помощью кодов, включающих номер версии указанных словарных элементов. В примере выше для облегчения восприятия описание представляется предпочтительными именами класса и свойства.

Примечание 2 — В рамках рассматриваемого онтологического подхода каждый продукт представляется как экземпляр в рамках данной онтологии.

Фундаментальные допущения используемого метода кодирования:

- в процессе обмена как отправитель, так и получатель должны ассоциировать один и тот же смысл с одной и той же характеристикой;
- характеристика, записанная в момент времени **time = t,** должна интерпретироваться с тем же смыслом в момент времени **time = t + 1**, даже если ссылочный словарь изменяется в промежутке между моментами времени **t** и **t + 1**.

В общем случае имеются два решения, учитывающих оба фундаментальных допущения:

- если ссылочный словарь меняется в промежутке между моментами **t** и **t + 1** и на допустимые изменения ограничения не накладываются, то пользователь ссылочного словаря должен иметь доступ

к ссылочному словарю как в момент времени **t**, так и в момент времени **t + 1**, т. е. ко всем различным версиям ссылочного словаря;

- другое решение определяется правилами управления изменением словаря, определенными в настоящем стандарте. Оно позволяет использовать только одну версию ссылочного словаря, а именно — более позднюю версию всех словарных элементов.

Данное решение, называемое *принципом онтологической непрерывности,* ограничивает допустимые изменения теми, которые гарантируют, что любая характеристика продукции, определенная в момент времени **time = t,** со ссылочным словарем, существующим в данный момент времени, сохраняем свой смысл, если она интерпретируется ссылочным словарем, существующим в момент времени **time = t + 1**. Следовательно, смысл словарного элемента, введенного в некоторый момент времени, будет сохранен в будущем.

Примечание 3 — В течение всего времени жизни описание ссылочного словаря может содержать небольшие ошибки («опечатки»). Словарь также нужно постоянно редактировать, чтобы отслеживать технологические усовершенствования. И, наконец, определения ссылочного словаря могут содержать концептуальные ошибки, когда следует изменять смысл класса и/или свойства.

Правила управления изменением словаря, установленные в настоящем разделе, классифицируют различные изменения, потребность в которых возникает в течение срока службы ссылочных словарей. В настоящем разделе также установлен порядок внесения изменений при условии, что смысл рассматриваемого понятия сохраняется, и он ассоциируется с существующей характеристикой.

9.2 Пересмотры и версии

Эффективность вносимого изменения зависит от его влияния на существующие и будущие характеристики. Сначала мы определяем, что означает тот факт, что характеристика соответствует ссылочному словарю.

Пусть:

- O_t — это ссылочный словарь O в момент времени **time = t**;
- C_t — классы ссылочного словаря O в момент времени **time = t**;
- P_t — свойства ссылочного словаря O в момент времени **time = t**;
- **applicable_properties$_t$** — функции, которые ставят в соответствие каждому классу C_t его применимые свойства из P_t в момент времени **time = t;**

Примечание 1 — Сущность **applicable_properties$_t$** (**Class_c$_t$**) включает все свойства, объявленные атрибутом **described_by** сущности **class**, определяющей сущность **Class_c$_t$**, если сущность **Class_c$_t$** представляет собой условный класс и свойства, импортированные сущностью **imported_properties**, входящие в класс **Class_c$_t$** — все применимые свойства допустимых суперклассов для класса **Class_c$_t$**.

- **domain$_t$** — функция, которая ставит в соответствие каждому свойству Pi_t в P_t его область значений в момент времени **time = t**.

Примечание 2 — Сущность **domain$_t$**(Pi_t) представляет собой область значений, объявленную атрибутом **domain** сущности **property_det**, определяющей свойство Pi в момент времени **time = t**.

Характеристика x_t удовлетворяет требованиям ссылочного словаря O_t, если и только если x_t может быть представлена как реализация O_t. Это означает, что:

- x_t принадлежит одному классу C_t, например, **Class_c$_t$**;
- x_t характеризуется значениями нескольких свойств, например, $P1_t, P2_t, \ldots, Pn_t$;
- $P1_t, P2_t, \ldots, Pn_t$ принадлежат P_t;
- $P1_t, P2_t, \ldots, Pn_t$ — свойства, применимые в классе **Class_c$_t$;**

Примечание 3 — Свойства $P1_t, P2_t, \ldots, Pn_t$ могут быть набором всех свойств, применимых в классе **Class_c$_t$**. Они также могут быть подмножеством данного множества.

- для каждого свойства $P1_t, P2_t, \ldots, Pn_t$ значение, назначенное данному свойству, принадлежит области значений данного свойства в момент времени **time = t** по определению функции **domain$_t$**(Pi_t), $i = 1, \ldots, n$.

Формально x_t удовлетворяет требованиям ссылочного словаря O_t, если пользователь словаря может записать это в виде:

$$x_t = \text{Class_c}_t(P1_t = v1, P2_t = v2, \ldots, Pn_t = vn)$$

При условии:

$$Class_c_t \in O_t, P1_t \in O_t, P2_t \in O_t, \ldots, Pn_t \in O_t$$

$$\wedge P1_t \in applicable_properties_t(Class_C_t) \wedge P2_t \in applicable_properties_t(Class_C_t)$$

$$\wedge \ldots \wedge Pn_t \in applicable_properties_t(Class_c_t)$$

$$\wedge v1 \in domain_t(P1_t) \wedge v2 \in domain_t(P2_t) \wedge \ldots \wedge vn \in domain_t(Pn_t)$$

Множество всех характеристик x_t, удовлетворяющих требованиям словаря O_t, называется *совокупностью* Pop_t O_t. Определение совокупности:

Pop_t = все x_t, удовлетворяющие требованиям O_t

Будем говорить, что:

- Pop_t и x_t *удовлетворяют требованиям* O_t;
- O_t *интерпретирует* Pop_t и x_t.

Данные определения позволяют классифицировать различные изменения ссылочных словарей.

Первый вид изменений в ссылочном словаре — это те изменения, которые никак не модифицируют множество характеристик, определенных данным ссылочным словарем, т. е. рассматриваемую совокупность ссылочного словаря (например, когда исправлена «опечатка», когда добавлены новые переводы, когда определение класса рассматривается повторно, чтобы сделать формулировку более ясной без изменения ее смысла). В данном случае сущность Pop_t (совокупность O_t в момент времени **time = t**) идентична сущности Pop_{t+1} (совокупности O_{t+1} в момент времени **time = t + 1**). Это означает, что:

- любая характеристика x_t, определенная для O_t, также *удовлетворяет требованиям* O_{t+1}. Таким образом, O_{t+1} *обратно совместим* с O_t, так как это позволяет *интерпретировать* все реализации;
- любая характеристика x_{t+1}, определенная для O_{t+1}, также *удовлетворяет требованиям* O_t. Таким образом, O_{t+1} также *совместим снизу-вверх* с O_t, так как O_t позволяет *интерпретировать* все реализации O_{t+1}.

В случае изменений, для которых имеет место «обратная» совместимость и совместимость «снизу-вверх», нет необходимости фиксировать, что характеристика «x» была построена в момент времени **time = t** или в момент времени **time = t + 1**. Таким образом, данное изменение не требует изменения № версии, ранее назначенного для нескольких словарных элементов в момент времени **time = t**. Данное изменение называется *изменением пересмотра.* Оно отслеживается либо путем увеличения атрибута **revision** словарного элемента (модифицируемого, когда рассматриваемое изменение искажает описание на языке источника, на котором дано определение словарного элемента), либо увеличением одного или нескольких атрибутов **translation_revision,** соответствующих другим языкам, на которые данный словарный элемент переведен, если рассматриваемое изменение искажает описание на соответствующем языке.

Примечание 4 — Номера пересмотров не регистрируются в идентификаторах словарных элементов. Характеристика, использующая рассматриваемые словарные элементы, позволяет использовать для интерпретации как O_t, так и O_{t+1}.

Примечание 5 — Каждый словарный элемент имеет атрибут пересмотра. Каждый переведенный словарный элемент имеет административный элемент данных **administrative_element**, описывающий исходный язык **source_language**, на котором словарный элемент **dictionary_element** был изначально определен, и атрибут пересмотра перевода **translation_revision** для каждого перевода.

Второй вид изменений в ссылочном словаре — это те изменения, которые уточняют ссылочный словарь и позволяют определить новые характеристики. Вводятся новые классы и новые свойства, добавляются новые значения свойств в область их значений. В соответствии с принципом онтологической непрерывности нельзя удалять классы, свойства и значения. Ссылочный словарь O_{t+1}, определенный уже после внесения изменений, должен быть в состоянии интерпретировать совокупность Pop_t. Словарь O_{t+1} по-прежнему *обратно совместим* со словарем O_t. Он позволяет интерпретировать все реализации. Но он уже не *совместим снизу-вверх,* так как некоторые характеристики, удовлетворяющие требованиям O_{t+1}, не удовлетворяют требованиям O_t.

В случае изменений, для которых справедлива только обратная совместимость, характеристика «x» (построенная в момент времени **time = t + 1** и зависящая от модифицированных словарных элементов) должна четко отражать это в своем представлении. Указанное изменение называется *изменением версии.* Оно отслеживается путем увеличения № версии словарного элемента, который был модифицирован, и №№ версий всех других словарных элементов, которые так же были модифицированы, как следствие.

Примечание 6 — № версий регистрируются в идентификаторах словарных элементов. Указание № версии элемента в его характеристике препятствует использованию словаря O_t для интерпретации характеристик, основанных на отличных версиях словаря O_{t+1}.

Таблица 1 суммирует отличия между версиями и пересмотрами.

Таблица 1 — Пересмотр и версия

Наименование	Обратная совместимость Совокупность Pop_t удовлетворяет требованиям O_{t+1}	Совместимость снизу-вверх Совокупность Pop_{t+1} удовлетворяет требованиям O_t
Пересмотр	Да	Да
Версия	Да	Нет

9.3 Исправление ошибок

При наличии ошибок в ссылочном словаре, ранее использованном для определения характеристик продукции, необходимо их исправить. Необходимо также определить механизм, позволяющий пользователям ссылочного словаря понимать сам процесс исправления ошибок. Для каждого набора данных, содержащего характеристики продукции, обработка ошибок означает: (1) распознавание, какая характеристика содержит ошибку, (2) определение порядка исправления ошибочных характеристик в соответствии с требованиями ссылочного словаря.

Если ошибочные словарные элементы еще не использовались для создания характеристик продукции (в замкнутой среде пользователя), то характеристики могут исправляться вместе со ссылочным словарем. Поставщик ссылочного словаря устанавливает порядок удаления ошибочных элементов из рассматриваемого ссылочного словаря, а также порядок внесения исправлений в сам ссылочный словарь.

Правила управления изменением словаря, определенные в настоящей части ИСО 13584, принимают, что среда пользователя является открытой. Это означает, что не все возможные характеристики доступны поставщику словаря, что указанные исправления не могут выполняться вместе с исправлениями ссылочного словаря. В такой среде следует использовать механизм «нецелесообразности использования».

«Нецелесообразность использования» означает, что:

- чтобы гарантировать обратную совместимость, ошибочные словарные элементы (значения свойств) остаются в ссылочном словаре, но при этом:

- все ошибочные элементы ассоциируются с атрибутом **is_deprecated,** имеющим значение **true,** что означает: «данный словарный элемент (значение свойства) не рекомендуется использовать далее в новых условиях», а также;

- атрибут, ассоциированный с каждым атрибутом **is_deprecated**, называется атрибутом интерпретации нецелесообразности использования **is_deprecated_interpretation.** Данный атрибут определяет порядок изменения характеристики, ссылающейся на не рекомендуемый для использования элемент, в соответствии с обновляемым ссылочным словарем.

Примечание 1 — Спецификация атрибута **is_deprecated_interpretation** может быть неформальной (объяснение пользователю ссылочного словаря порядка обработки соответствующих данных) или формальной (инструкция по настройке компьютера для автоматического исправления данных).

Примечание 2 — В рассматриваемой спецификации правил управления изменением словаря нет формального языка представления содержания атрибута **is_deprecated_interpretation**. В настоящий момент команда-разработчик общей словарной модели ИСО 13584/МЭК 61360 разрабатывает такой язык.

Пример 1 — Если в классе C1 применимое свойство P1, значение которого выражается в метрах, заменяется свойством P2, имеющим тот же смысл, но выражаемым в микронах, то: (1) P1 значение атрибута is_deprecated равно true, (2) атрибут is_deprecated_interpretation может иметь значение: «Значение данного свойства должно теперь выражаться в микронах и относиться к свойству P2».

Пример 2 — В вышеприведенном Примере 1 значение атрибута свойства P1 is_deprecated_interpretation может быть представлено (если используемый подход согласован с сообществом, использующим рассматриваемый ссылочный словарь) как выражение, использующее синтаксис языка EXPRESS и представляющее значения свойств через идентификаторы свойств. В данном случае выражение имеет вид: «P2:= P1 * 1 000 000».

На рисунке 1 приведена модель планирования, представляющая указанный механизм в общей словарной модели ИСО 13584/МЭК 61360, вместе с определениями соответствующих атрибутов.

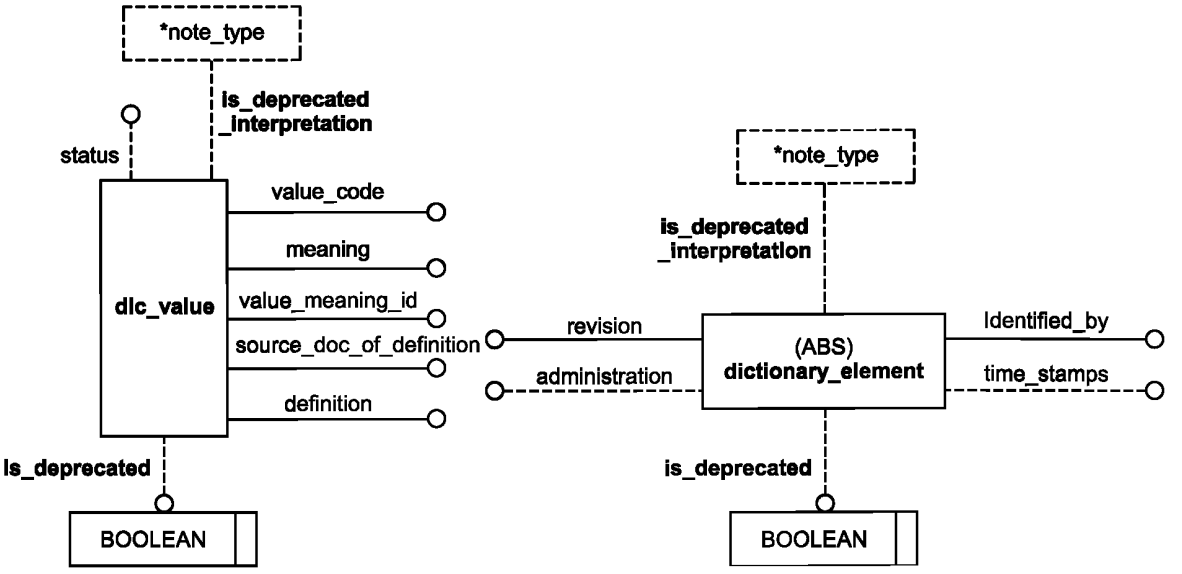


Рисунок 1 — Информационная модель элементов, не рекомендуемых для использования

note_value	Тип примечания
is_deprecated_interpretation	Интерпретация нецелесообразности использования элемента
status	Статус
value_code	Код значения
meaning	Смысл
value_meaning_id	Идентификатор смысла значения
dic_value	Словарное значение
source_doc_of_definition	Исходный документ для формулировки определения
definition	Определение
boolean	Булев тип
Is_deprecated	Не рекомендуется для использования
revision	Пересмотр
administration	Административные данные
(ABS) dictionary_element	Словарный элемент (в алфавитном порядке)
Identified_by	Идентифицирован ...
time_stamps	Временные отметки

Определения атрибутов:

is_deprecated: по выбору, Булевский тип. Если атрибут равен **true**, то элемент **dic_values/dictionary_element** не должен далее использоваться.

is_deprecated_interpretation: интерпретирует нецелесообразность использования элемента, устанавливает порядок интерпретации реализаций значений, не рекомендуемых для использования элементов.

9.4 Правила управления изменениями

Данный подраздел устанавливает правила внесения изменений в ссылочный словарь.

9.4.1 Критерии для классификации изменений

Оценка влияния изменения словарного элемента на совокупности характеристик, интерпретируемых словарями O_t и/или O_{t+1}, дает критерии для классификации влияния изменений, таких как изменение номера пересмотра, изменение номера версии или исправление ошибок. В данном разделе показано, как нужно регистрировать каждое изменение в соответствии с его влиянием, чтобы гаранти-

ровать, что получатель файла обмена, содержащего характеристики элемента, поймет, позволяет ли имеющийся словарь интерпретировать данный файл обмена или нет.

Рассматриваемые правила определяют минимальные требования. При этом поставщик ссылочного словаря всегда может принять решение: 1) обновить номер версии словарного элемента, тогда как правила запрашивают только обновление номера пересмотра, 2) не рекомендовать для дальнейшего использования модифицированный элемент, тогда как правила запрашивают только обновление номера пересмотра или номера версии.

Правило 1: Изменение номера пересмотра

Если после преобразования сущности $\mathbf{Ent}_t$ (класса, типа данных, свойства, имени, определения...) ссылочного словаря O_t в сущность $\mathbf{Ent}_{t+1}$ (1), новый ссылочный словарь O_{t+1} может интерпретировать все характеристики, определяемые словарем O_t, и (2) не позволяет определять какие-либо новые характеристики, то рассматриваемое изменение представляет собой изменение номера пересмотра словарного элемента путем изменения сущности $\mathbf{Ent}_t$.

Если описание измененного словарного элемента определено только на одном языке или если оно уже переведено, а изменение искажает описание на исходном языке (на котором было дано определение), то рассматриваемое изменение должно увеличить значение атрибута **revision** словарного элемента, модифицированного указанным изменением. Если изменение сущности **Ent** искажает переводы на другие языки (на которые переведен словарный элемент), то соответствующие переводы также должны быть изменены. Также следует увеличить значения атрибутов **translation_revision** соответствующих переводов.

Пример 1 — Если словарь доступен только на одном языке, то (при изменении определения класса без изменения характеристик, которые он может интерпретировать) атрибут пересмотра класса должен быть увеличен.

Пример 2 — Если оригинальный язык словаря — английский, и словарь переведен на немецкий и французский, и если изменяется французское определение класса без изменения характеристик, которые данный класс может интерпретировать, то следует увеличить атрибут translation_revision французского перевода.

Пример 3 — Если оригинальный язык словаря — английский, и словарь переведен на немецкий и французский, и если изменяется значение атрибута figure класса без изменения характеристик, которые данный класс может интерпретировать, то атрибут revision класса следует увеличить. Если значением атрибута figure является graphic_files (данное значение является языково-независимым) и, таким образом, применяется для описаний на всех языках, то атрибуты translation_revision как немецкого, так и французского переводов можно не увеличивать, так как перевод не изменялся.

Пример 4 — Если в класс добавляется видимое свойство без обеспечения его применимости в данном классе или во всех его подклассах, то для указанного нового свойства нет описываемых характеристик. Никакие прямые атрибуты рассматриваемого класса не модифицируются, № пересмотра и № версий данного класса не требуют обновления.

Правило 2: Изменение номера версии

Если после замены сущности $\mathbf{Ent}_t$ (класса, типа данных, свойства,...) ссылочного словаря O_t на сущность $\mathbf{Ent}_{t+1}$ (1) новый ссылочный словарь O_{t+1} может интерпретировать все характеристики, определяемые ссылочным словарем O_t, но (2) новый словарь также дает новые характеристики, не интерпретируемые словарем O_t, то рассматриваемое изменение должно увеличивать номер версии $\mathbf{Ent}_t$.

Примечание 1 — Ограничения оказывают влияние на те характеристики элементов, которые удовлетворяют ограничениям. Таким образом, изменения ограничений должны представляться путем увеличения номера версии класса, содержащего данные ограничения. При этом изменение ограничения не изменяет множества характеристик, интерпретируемых словарем. Поэтому когда ограничения модифицируются в классе, рассматриваемое множество характеристик элементов, удовлетворяющих ограничениям, может расширяться или сужаться без нарушения принципа онтологической непрерывности.

Пример 5 — Если применимое свойство добавляется в класс, то (1) все характеристики, определенные предшествующим ссылочным словарем, могут по-прежнему им интерпретироваться (без использования нового применимого свойства), но при этом (2) новым ссылочным словарем могут быть определены некоторые характеристики (использующие новые применимые свойства), не интерпретируемые предшествующим словарем. Таким образом, номер версии класса должен увеличиться.

Пример 6 — Если новая альтернативная единица измерения добавляется к мере действительного типа некоторого свойства, то все характеристики, определенные предшествующим ссылочным словарем, могут по-прежнему интерпретироваться (без использования новой альтернативной единицы измерения). При этом некоторые новые характеристики (не интерпретируемые предшествующим

словарем) могут быть определены новым ссылочным словарем (путем использования новой альтернативной единицы измерения). Таким образом, № версии класса должен увеличиться.

Правило 3: Исправление ошибок

Если после замены словарного элемента $\mathbf{Ent_t}$ (класса, типа данных, свойства,...) ссылочного словаря на сущность $\mathbf{Ent_{t+1}}$ новый ссылочный словарь O_{t+1} уже не может интерпретировать все характеристики, определяемые словарем O_t, то рассматриваемое изменение не является обратно совместимым и нарушается принцип онтологической непрерывности.

Для исправления ошибок необходимо: (1) идентифицировать те словарные элементы, которые следует модифицировать, и назначить значение **true** их атрибуту **is_deprecated**; (2) определить новые сущности, которые будут исправлять ошибки; (3) описать в атрибуте **is_deprecated_interpretation** не рекомендуемый для использования элемент (почему данный элемент уже не рекомендуется для использования, как следует изменять характеристику, ссылающуюся на не рекомендуемый для использования элемент, в соответствии с требованиями обновляемого ссылочного словаря).

Пример 7 — Если вносится исправление единицы измерения меры действительного типа некоторого свойства, то рассматриваемая продукция, зарегистрированная ранее и характеризуемая данным свойством, должна описываться уже по-другому. Новый ссылочный словарь уже не является обратно совместимым, и данное изменение не может вноситься просто так. Поэтому (1) соответствующее свойство следует считать не рекомендуемым для использования, (2) новое свойство (уже с другим идентификатором) должно создаваться как видимое и применимое там, где таковым было предшествующее свойство, (3) в атрибуте is_deprecated_interpretation предшествующего свойства необходимо указать, что значение данного свойства должно быть, например, «... разделено на 1000. Только после этого оно может рассматриваться как значение нового свойства».

Пример 8 — Если некоторые новые контекстные параметры добавляются к тем, от которых зависит контекстно-зависимое свойство, то все характеристики, использующие данное свойство, должны уже описываться по-другому после внесения изменения. Новый ссылочный словарь уже не может интерпретировать некоторые предшествующие характеристики. Поэтому необходимо (1) сделать не рекомендуемым для использования старое контекстно-зависимое свойство, (2) создать и внести новое свойство, (3) объяснить нецелесообразность использования старого свойства и, возможно (если предшествующее контекстно-зависимое свойство было предназначено для измерения при фиксированном значении нового контекстного параметра), порядок преобразования значения не рекомендуемого для использования контекстно-зависимого свойства в значение нового свойства.

Примечание 2 — В данном примере также возможно сохранять в ссылочном словаре как предшествующее, так и новое контекстно-зависимое свойство.

На рисунке 2 приведена процедура классификации вносимых изменений.

Change request	Запрос на изменение
Backward compatible?	Является ли сущность обратно совместимой?
Yes	Да
No	Нет
Upward compatible?	Является ли сущность совместимой снизу-вверх?
Deprecate Ent_t Introduce new entities = Ent_{t+1}	Объявить сущность $\mathbf{Ent_t}$, не рекомендуемой для использования Внести новые сущности $\mathbf{Ent_{t+1}}$
Revision	Пересмотр
Version	Версия
Out of scope	Не рассматривается
Request an internal agreement of dictionary users	Запрос на внутреннее согласование пользователей словаря

9.4.2 Изменение зависимости и распространения

В ссылочном словаре каждый словарный элемент может существовать только в одной версии. Таким образом, если номер версии словарного элемента ссылочного словаря увеличивается, то все словарные элементы, ссылающиеся на данный словарный элемент, должны быть изменены, чтобы ссылаться уже на новую версию. В самом деле, такие изменения должны отслеживаться на уровне идентификаторов всех

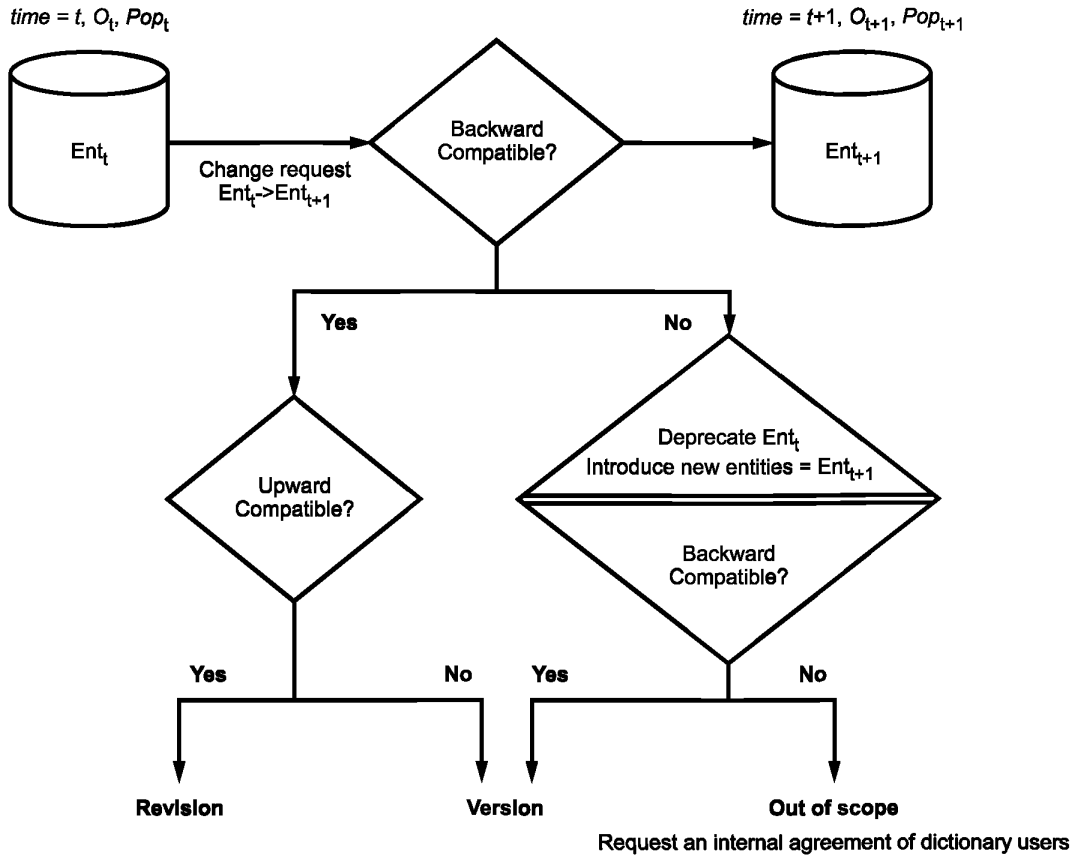


Рисунок 2 — Классификация изменений словаря

ссылочных словарных элементов, чтобы получить уверенность в том, что когда идентификатор словарного элемента заменяется соответствующим описанием словарного элемента, он содержит корректные внутренние ссылки. Таким образом, каждое изменение номера версии словарного элемента (на который производятся ссылки из другого словарного элемента) должно быть представлено новой версией последнего.

Исключительность каждого словарного элемента обеспечивается внутри одного ссылочного словаря, а не в нескольких словарях одновременно. Если класс C1 ссылочного словаря D1 ссылается на класс C2 ссылочного словаря D2 с помощью условного соотношения, то на поставщике ссылочного словаря D1 лежит ответственность за принятие решения: на какие классы производится ссылка, на какую версию, и какие свойства импортируются. Таким образом, если поставщик словаря D2 изменяет версию класса C2, то на поставщике словаря D1 лежит ответственность за принятие решения: при каких условиях и когда на новую версию класса C2 можно будет ссылаться из класса C1. Старые ссылки могут сохраняться. Но если № версии ссылки увеличивается, то № версии класса C1 также должен быть увеличен.

Вышесказанное суммируется в четырех правилах:

Правило 4: отсутствие распространения на ссылочные словари

Если класс C1 ссылочного словаря D1 ссылается на класс C2 ссылочного словаря D2 с помощью условного соотношения, и если № версии класса C2 обновлен, то на поставщике словаря D1 лежит ответственность за принятие решения: при каких условиях и когда класс C1 может ссылаться на новую версию класса C2. Если это имеет место, то № версии класса C1 следует увеличить.

Пример 1 — Класс C1 импортирует с помощью условного соотношения свойства P1 и P2. Новое применимое свойство P3 добавляется в класс C2. Поставщик словаря D1 не заинтересован в свойстве P3. Тогда класс C1 может продолжать ссылаться на предшествующую версию класса C2.

Правило 5: распространение версии

Увеличение номера версии любого словарного элемента, на который ссылаются другие словарные элементы того же ссылочного словаря, должно распространяться и на них.

Примечание 1 — Это те же средства ссылочного словаря, идентифицированные тем же поставщиком информации.

Примечание 2 — Если один словарный элемент ссылается на другой словарный элемент, то данная ссылка производится через базовую семантическую единицу BSU, включающую версию целевого словарного элемента. Таким образом, если версия целевого словарного элемента изменена, то содержание исходного словарного элемента также должно измениться для регистрации исправленной (новой) ссылки. Данное изменение означает, что новая характеристика может быть описана (косвенно) исходным словарным элементом, но его версия должна измениться.

Пример 2 — При изменении версии поименованного типа named_type (например, для расширения его области значений) необходимо также изменить область значений всех свойств, которые ссылаются на данный тип как на область domain. Таким образом, версию данных свойств также следует обновить. Также необходимо изменить множество характеристик, описываемых классами, в которых данные свойства описываются как применимые с помощью атрибута described_by.

Пример 3 — Изменение номера версии класса ведет к изменению номеров версий его подклассов, и так далее.

Пример 4 — Если номер версии класса определений свойств увеличивается, то номер версии данного свойства также должен быть увеличен.

Примечание 3 — Данное правило гарантирует, что если номер версии характеристического класса (используемый для характеристики элемента в файле обмена) меньше или равен соответствующему номеру версии в локальном словаре получающей системы, то данная система может корректно интерпретировать рассматриваемую характеристику любой сложности.

Правило 6: вычисление значения номера новой версии

Для каждого заданного изменения все распространяемые изменения должны рассматриваться вместе, и номер версии каждой сущности должен увеличиваться максимум один раз. Допускается сгруппировать несколько различных изменений, чтобы подсчитать номера новых версий множества словарных элементов.

Правило 7: Циркуляция новой версии

На поставщике ссылочного словаря лежит ответственность о принятии решения: как и когда следует распространять обновления.

Пример 5 — Ссылочный словарь может быть ассоциирован с сервером, соответствующим требованиям ИСО/ТС 29002-20 и делающим доступным каждое обновление, как только оно вступает в силу.

Пример 6 — Ссылочный словарь распространяется выпусками. Каждый год новый выпуск Release интегрирует все обновления, разработанные в течение года. В данном случае для всех модифицированных словарных элементов может быть увеличен номер только одной версии.

9.4.3 Управление классами категоризации

Классы категоризации не оказывают влияние на характеристики элементов. На них вышеуказанные правила не распространяются.

Правило 8: разработка версий классов категоризации

Увеличение номера версии классов категоризации необходимо, если изменяется один или несколько их суперклассов, на которые производится ссылка атрибута **categorization_class_superclasses**. Все другие изменения регистрируются путем увеличения номера пересмотра.

Изменение номера версии классов категоризации не распространяется на характеристические классы, которые на них ссылаются. Указанные изменения регистрируются только путем изменения номера пересмотра.

9.4.4 Управление номером версии и номером пересмотра словаря

Во время обмена файлов (словарных характеристик элементов) нижеследующие правила гарантируют, что в процессе обмена (просто путем проверки версии словаря) получатель файла может узнать, можно ли с помощью доступной версии словаря интерпретировать рассматриваемый файл.

Правило 9: версии и пересмотры словаря

Если обновленный словарь распределяется в соответствии с Правилом 6, то, в этом случае:

- если номер рассматриваемой версии **version** какого-либо словарного элемента, определенного в данном словаре, увеличен, и/или если внесен новый словарный элемент, то номер версии словаря также должен быть увеличен;

Пример — Новый словарный элемент вносится в словарь, если внесен новый подкласс существующего класса или определен новый видимый тип.

- если номер пересмотра какого-либо словарного элемента, определенного в данном словаре, увеличен, а номер версии словаря не изменен, то номер пересмотра словаря должен быть увеличен;
- если номер версии словаря увеличен, то номер пересмотра должен быть поставлен на '0'.

9.5 Изменения и атрибуты словаря

9.5.1 Атрибуты, поддерживаемые системой

Правила управления изменением словаря ограничены атрибутами, доступными для изменений, производимыми по запросам пользователей. Атрибуты, поддерживаемые системой, не рассматриваются при изменениях ссылочного словаря, так как они модифицируются автоматически как следствие других изменений:

- *Изменение номера пересмотра*: если происходит изменение описания на исходном языке (на котором описан словарный элемент), то номер пересмотра увеличивается, а дата пересмотра **date_of_current_revision** обновляется. Если происходит изменение описания на одном или нескольких других языках (на которые переведено описание словарного элемента), то номер пересмотра перевода **translation_revision** на соответствующий язык увеличивается, а дата пересмотра перевода **date_of_current_translation_revision** на соответствующий язык обновляется;
- *Изменение номера версии*: номер версии увеличивается и дата текущей версии **date_of_current_version** обновляется с течением времени. Номер пересмотра устанавливается на минимальное значение атрибута пересмотра, а дата текущего пересмотра **date_of_current_revision** обновляется;
- *Создание нового словарного элемента*: новый словарный элемент создается с новым кодом, а дата исходного определения **date_of_original_definition** устанавливается по текущему времени. Номер версии устанавливается на минимальное значение, определенное для атрибута версии, а дата текущей версии **date_of_current_version** обновляется по текущему времени. Номер пересмотра устанавливается на минимальное значение, а дата текущего пересмотра **date_of_current_revision** обновляется по текущему времени.

9.5.2 Атрибуты, доступные для текстовых изменений

Роль терминологических атрибутов словарных элементов (имени, определения, примечания, иконки и т. п.) заключается в том, чтобы объяснить, какой вид продукции характеризуется заданным классом ссылочных словарей и какой вид характеристик продукции представляется каждым отдельным свойством.

Для обеспечения обратной совместимости текстовых изменений терминологических атрибутов рассматриваемые изменения не должны искажать смысл класса (свойства), даже если данные изменения уточняют его смысл. Вместе с тем, текстовые изменения могут расширить класс определений, например, путем принятия во внимание разработки новой продукции.

Таким образом, текстовые изменения требуют, по крайней мере, задания нового № пересмотра. На поставщике словаря лежит ответственность о принятии решения: следует ли вместе с вносимым изменением менять версию словарного элемента, чтобы гарантировать, что пользователи словаря получат доступ к терминологическим атрибутам, используемым при определении некоторых характеристик. На поставщике словаря также лежит ответственность за принятие решения: следует ли ассоциировать новый определяемый словарный элемент с новым кодом, так как новое определение может не быть обратно совместимым с предшествующим определением.

9.6 Ограничения на эволюцию ссылочных словарей

В данном разделе суммируются ограничения для каждого вида понятий (классов, соотношений между классами, свойств и характеристик) в процессе эволюции ссылочных словарей.

Неизменность классов

Существование характеристического класса нельзя отрицать в процессе эволюции. Дело в том, что каждый характеристический класс позволяет определить некоторые характеристики, любой класс, существующий в момент времени **time = t**, продолжает существовать в момент $\mathbf{t'}$ при $\mathbf{t' > t}$.

Примечание 1 — Чтобы сделать модель управления изменениями более гибкой, предполагается, что класс может устаревать. При этом он маркируется как «не рекомендуемый для использования», может быть заменен другим классом и продолжает принадлежать новым версиям ссылочного словаря.

Указанный принцип говорит, что новейшая версия ссылочного словаря способна интерпретировать все ранее определенные характеристики. На каждом пользователе ссылочного словаря лежит ответственность за принятие решения: при каких условиях и до какого момента времени не рекомендуемые для использования элементы должны храниться в словаре пользователя.

Проблема неизменности имеет особенности для классов категоризации. Классы категоризации не используются при определении характеристик, поэтому данные классы могут быть погашены (модифицированы) без создания проблемы обеспечения обратной совместимости.

Неизменность свойств

Аналогично, все свойства, существующие в момент времени **time = t,** должны по-прежнему существовать в момент **t′** при **t′ > t**. Свойства также могут устаревать. Но при этом ни условия существования, ни значения заданного элемента не могут изменяться. Область значений свойств может участвовать в процессе эволюции. Принимая во внимание требование обеспечения обратной совместимости, область значений может только увеличиваться. При этом отдельные значения время от времени маркируются как не рекомендуемые для использования.

Неизменность соотношения «класс-подкласс»

Соотношения типа «класс-подкласс» — это соотношения между классами и всеми подклассами. Эти соотношения — прямые или удовлетворяющие свойству транзитивности. Соотношение «класс-подкласс» поддерживает наследственность между суперклассом и подклассом. Выполнение требования неизменности заданного соотношения «класс-подкласс» между двумя классами С1 (суперклассом) и С2 (подклассом) зависит от последствий реализации данного соотношения для характеристик, определенных подклассом:

- если С2 не наследует от С1 какой-либо элемент (свойство, тип, значение,...), используемый в характеристике, то соотношение «С1-С2» может быть погашено. Последствия реализации данного соотношения для № версии и № пересмотра определяются правилами управления изменением словаря;
- если С2 наследует от С1 некоторый элемент (свойство, тип, значение,...), используемый в реализации характеристики С2, то соотношение «С1-С2» погашено быть не может.

Отметим, что данное ограничение допускает существенную эволюцию иерархии соотношений «класс-подкласс», например, допускает включение промежуточных классов между двумя классами, связанными соотношением «класс-подкласс».

Неизменность характеристик

Из факта, что свойство Р применимо в классе С в момент времени **time = t,** следует, что Р остается применимым в С в момент **t′** при **t′ > t**.

Примечание 2 — Для этого совсем не требуется, чтобы одни и те же применимые свойства всегда использовались для описания экземпляров одного и того же класса. Свойства, используемые для характеристики элемента, не зависят от эволюций ссылочного словаря. Они зависят, в основном, от требований приложения, использующего ссылочный словарь.

Примечание 3 — Если свойство Р1 объявлено как применимое в классе С2, являющимся подклассом С1, то Р1 может стать применимым в классе С1 без возникновения проблемы обеспечения обратной совместимости, так как свойство применимости является унаследованным.

Приложение А
(обязательное)

Обзор кодов классификации типов неколичественных типов элементов данных (основной класс А)

Таблица А.1 — Обзор кодов классификации типов неколичественных типов элементов данных (основной класс А)

Коды классификации типов	Описание
А11	Географическая единица (больше чем населенный пункт)
А12	Географическое размещение (населенный пункт и менее)
А13	Географический маршрут или сеть
А21	Организация
А22	Должностное лицо
А31	Срок, заданный датой и временем
А32	Время дня
А41	Частное лицо
А51	Продукт
А52	Класс продуктов
А53	Партия или упаковка продукта (тип)
А54	Способ, средства и единицы транспортировки
А55	Процесс производства и технология
А56	Функция и приложение продукта
А57	Материал
А58	Геометрия, формы и размеры продукта
А59	Качество, функционирование и испытания продукта
А61	Документ и сообщение
А62	Элемент информации и информационная группа
А63	Среда и единицы передачи данных
А71	Единицы измерения
А79	Тип измерения
А81	Расчет
А82	Проект, проектное мероприятие
А83	Процедура
А91	Абстрактная идентификация (язык, цвет, …)
А93	Раздел

Приложение В
(обязательное)

Сущность короткого имени

В Таблице В.1 даны короткие имена, рассмотренные в настоящей части ИСО 13584. Требования к коротким именам определены методами практической реализации, рассмотренными в ИСО 10303.

Таблица В.1 — Сущность короткого имени

Полное имя	Сокращенное имя
A_PRIORI_SEMANTIC_RELATIONSHIP	APS1
ADMINISTRATIVE_DATA	ADMDT
AXIS1_PLACEMENT_TYPE	AXPLTY
AXIS2_PLACEMENT_2D_TYPE	AP2T
AXIS2_PLACEMENT_3D_TYPE	AP3T
BASIC_SEMANTIC_UNIT	BSSMUN
BOOLEAN_TYPE	BLNTYP
CARDINALITY_CONSTRAINT	CRDCNS
CATEGORIZATION_CLASS	CTGCLS
CLASS	CLASS
CLASS_AND_PROPERTY_ELEMENTS	CAPE
CLASS_BSU	CLSBS
CLASS_BSU_RELATIONSHIP	CLBSRL
CLASS_CONSTRAINT	CLSCNS
CLASS_REFERENCE_TYPE	CLRFTY
CLASS_RELATED_BSU	CLRLBS
CLASS_VALUE_ASSIGNMENT	CLVLAS
COMPLEX_TYPE	CMPTYP
CONDITION_DET	CNDDT
CONFIGURATION_CONTROL_CONSTRAINT	CNCNCN
CONSTRAINT	CNSTRN
CONTENT_ITEM	CNTITM
CONTEXT_RESTRICTION_CONSTRAINT	CNRSCN
DATA_TYPE	DTTYP
DATA_TYPE_BSU	DTTYBS
DATA_TYPE_ELEMENT	DTTYEL
DATE_DATA_TYPE	DTDTTY
DATE_TIME_DATA_TYPE	DTDT
DATES	DATES

Продолжение таблицы В.1

Полное имя	Сокращенное имя
DEPENDENT_P_DET	DPPDT
DIC_UNIT	DCUNT
DIC_VALUE	DCVL
DICTIONARY_ELEMENT	DCTELM
DICTIONARY_IDENTIFICATION	DCTIDN
DOCUMENT	DCMNT
DOMAIN_CONSTRAINT	DMNCNS
ENTITY_INSTANCE_TYPE	ENINTY
ENTITY_SUBTYPE_CONSTRAINT	ENSBCN
ENUMERATION_CONSTRAINT	ENMCNS
EXTERNAL_GRAPHICS	EXTGRP
FILTER	FILTER
GLOBAL_LANGUAGE_ASSIGNMENT	GLLNAS
GRAPHIC_FILES	GRPFLS
GRAPHICS	GRPHCS
IDENTIFIED_DOCUMENT	IDNDCM
INT_CURRENCY_TYPE	INCRTY
INT_MEASURE_TYPE	INMSTY
INT_TYPE	INTTYP
INTEGRITY_CONSTRAINT	INTCNS
ITEM_CLASS	ITMCLS
ITEM_CLASS_CASE_OF	ICCO
ITEM_NAMES	ITMNMS
LABEL_WITH_LANGUAGE	LBWTLN
LANGUAGE_CODE	LNGCD
LEVEL_TYPE	LVLTYP
MATHEMATICAL_STRING	MTHSTR
NAMED_TYPE	NMDTYP
NON_DEPENDENT_P_DET	NDPD
NON_QUANTITATIVE_CODE_TYPE	NQCT
NON_QUANTITATIVE_INT_TYPE	NQIT
NON_SI_UNIT	NNSUN
NON_TRANSLATABLE_STRING_TYPE	NTST
NUMBER_TYPE	NMBTYP
PLACEMENT_TYPE	PLCTYP
PRESENT_TRANSLATIONS	PRSTRN

Окончание таблицы В.1

Полное имя	Сокращенное имя
PROPERTY_BSU	PRPBS
PROPERTY_CONSTRAINT	PRPCNS
PROPERTY_DET	PRPDT
RANGE_CONSTRAINT	RNGCNS
RATIONAL_MEASURE_TYPE	RTMSTY
RATIONAL_TYPE	RTNTYP
REAL_CURRENCY_TYPE	RLCRTY
REAL_MEASURE_TYPE	RLMSTY
REAL_TYPE	RLTYP
SIMPLE_TYPE	SMPTYP
STRING_PATTERN_CONSTRAINT	STPTCN
STRING_SIZE_CONSTRAINT	STSZCN
STRING_TYPE	STRTYP
SUBCLASS_CONSTRAINT	SBCCNS
SUPPLIER_BSU	SPPBS
SUPPLIER_BSU_RELATIONSHIP	SPBSRL
SUPPLIER_ELEMENT	SPPELM
SUPPLIER_RELATED_BSU	SPRLBS
TIME_DATA_TYPE	TMDTTY
TRANSLATABLE_STRING_TYPE	TRSTTY
TRANSLATED_LABEL	TRNLBL
TRANSLATED_TEXT	TRNTXT
TRANSLATION_DATA	TRNDT
URI_TYPE	URTYP
VALUE_DOMAIN	VLDMN

Приложение С
(обязательное)

Компьютерно-интерпретируемые листинги

Данное приложение ссылается на листинги множества схем языка EXPRESS, задокументированных в настоящей части ИСО 13584. Указанные схемы (без комментариев или пояснительных текстов) — это ресурсы других частей ИСО 13584. Приложение также содержит схемы длинной формы из указанного множества схем. Данная схема длинной формы, называемая **ISO13584_P42_2_LONG_FORM_SCHEMA**, нужна для описания и обмена простыми словарями. Всю указанную схематику смотри на сайте:

http://стандарты.iso.org/iso/13584/–42/EXPRESS

Если имеются трудности с получением доступа к данному сайту, обращайтесь прямо в Центральный секретариат ИСО или в секретариат ИСО/ТК184/ПК4 по электронной почте: gost.e-ecolog.ru

Примечание — ИСО 13584-32, известный как язык OntoML, является схемой языка расширенной разметки XML, основанной на механизмах, определенных в настоящей части ИСО 13584. Данный Стандарт обеспечивает возможности обмена более сложных словарей, требующих использования совместно-значимых свойств, функциональных моделей и подходов.

Нижеследующее замечание рассматривает компьютерно-интерпретируемые файлы.

В таблице С.1 ниже даны единые идентификаторы ресурса URI. Их компьютерно-интерпретируемые листинги даны в рассматривамом приложении.

Таблица С.1 — Схемы языка EXPRESS, задокументированные в настоящей части ИСО 13584

Описание	Единый идентификатор ресурса URI
ISO13584_IEC61360_dictionary_SCHEMA	urn:iso:std:iso:13584: — 42:ed-2: tech:express:dictionary
ISO13584_IEC61360_language_resource_SCHEMA	urn:iso:std:iso:13584: — 42:ed-2: tech:express:language
ISO13584_IEC61360_class_constraint_SCHEMA	urn:iso:std:iso:13584: — 42:ed-2: tech:express:constraint
ISO13584_IEC61360_item_class_case_of_SCHEMA	urn:iso:std:iso:13584: — 42:ed-2: tech:express:caseof
ISO13584_P42_2_LONG_FORM_SCHEMA	urn:iso:std:iso:13584: — 42:ed-2: tech:express:dictionary-long-form

Примечание — Для поддержки практической реализации может быть предоставлена дополнительная информация. Она дается единым указателем ресурса URL в приложении К.

Приложение D
(обязательное)

Спецификация формата значения

Настоящий стандарт и МЭК 61360-2 приводят конкретный синтаксис формата строк и численных значений, ассоциируемых со свойством.

Пример 1 — Формат NR1 3 устанавливает, что допустимы только целые, состоящие из трех цифр.

Примечание 1 — Формат значения типа данных data_type, включая булевский тип **boolean_type**, не устанавливается.

Примечание 2 — В настоящем стандарте определение формата значения свойства не обязательно.

Синтаксис допустимого формата определен в данном Приложении с помощью подмножества расширенной формы Бэкуса-Наура (EBNF), определенного в ИСО/МЭК 14977.

Пример 2 — Синтаксис формата NR1 3 — это буквы 'NR1' '' '3'.

Смысл каждого варианта синтаксиса (символов, используемых для представления значения) не может быть задан с помощью формы EBNF. Смысл каждой части формата символов, допустимых для представления значений, устанавливается отдельно для каждой части формата.

Пример 3 — Синтаксис формата NR1 3 имеет следующий смысл: NR1 означает, что можно представить только целое значение. Пробел означает, что можно использовать только фиксированное число символов. Цифра 3 означает допустимое количество цифр в записи.

D.1 Обозначения

Таблица D.1 содержит подмножество синтаксического мета-языка EBNF, определенного в ИСО/МЭК 14977 и использованного в настоящем стандарте. Данный язык устанавливает формат значений свойств.

С помощью введенных обозначений синтаксис рассматриваемого подмножества мета-языка EBNF (используемый в настоящем стандарте для задания формата значений свойств) характеризуется нижеследующей грамматикой (описание символов мета-идентификаторов, букв и цифр отсутствует):

```
syntax = syntaxrule,{ syntaxrule);
syntaxrule = metaidentifier, '=' definitionslist, '; ';
definitionslist = singledefinition, {'|', singledefinition};
singledefinition = term, {',', term};
term = primary, ('-', primary);
primary = optionalsequence | repeatedsequence | groupedsequence | metaidentifier | terminal | empty;
optionalsequence = ' ['definitionslist ']';
repeatedsequence = '{'definitionslist '}';
groupedsequence = '('definitionslist ')';
metaidentifier = letter, {letter];
terminal = «'», (character — «'»), {character — «'»}, «'»
| '»', (character — '»'),{character — '»'}, '»';
empty =;
```

Знак равенства '=' указывает синтаксическое правило. Мета-идентификатор слева можно заменить комбинацией элементов справа. Любые пробелы между элементами информации не несут, если только они не находятся внутри терминала. Синтаксическое правило закачивается символом ';'.

Таблица D.1 — Синтаксический мета-язык EBNF по ИСО/МЭК 14977

Представление	Названия символов по ИСО/МЭК 10646-1	Символ мета-языка и его роль
'	апостроф	Одинарная кавычка представляет терминал языка. Терминал не должен содержать апостроф. Пример: 'Hello'
«...»	кавычки	Двойная кавычка представляет терминал языка. Терминал не должен содержать кавычки. Пример: «Машина Джона»
( )	левая скобка, правая скобка	Символы начала/окончания группы. Содержание рассматривается как один символ

Окончание таблицы D.1

Представление	Названия символов по ИСО/МЭК 10646-1	Символ мета-языка и его роль
[ ]	левая квадратная скобка, правая квадратная скобка	Символ начала/окончания опции. Содержание может быть и может не быть
{ }	левая фигурная скобка, правая фигурная скобка	Символ начала/окончания повтора. Содержание может повторяться от 0 до *n* раз
–	тире-минус	Символ замены
,	запятая	Символ последовательного соединения
=	знак равенства	Символ определения. Синтаксическое правило: определяет символ слева через формулу справа
|	вертикальная линия	Альтернативный разделитель
;	точка с запятой	Символ терминатора. Окончание синтаксического правила

Использование мета-идентификатора внутри списка определений означает, что нетерминальный символ находится слева от другого синтаксического правила. Мета-идентификатор составлен из букв и цифр. Первый символ — буква. Если термин содержит и первичное выражение перед знаком «минус», и первичное выражение за знаком «минус», то только последовательность символов, представленная первым первичным выражением и не представленная вторым первичным выражением, представляется термином.

Пример 1 — Обозначение:

'''', символ-'''', ''''

означает любой символ (кроме апострофа), вставленный между двумя апострофами.

Терминал означает символ, который не может быть расширен далее с помощью синтаксического правила, и который появится в окончательном результате. Терминал может быть представлен двумя способами: либо это набор символов без апострофа (вставленный между двумя апострофами), либо это набор символов без кавычек (вставленный между двумя кавычками).

Пример 2 — Предположим, что мы хотим установить (с помощью данной грамматики) цену продукта в евро €. Цена — это положительное число с не более, чем 2 цифрами после запятой (центы). Определим три мета-идентификатора, ассоциированных с тремя синтаксическими правилами:

digit = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';

cents = ['.', digit [, digit]];

euros = digit (, digit) cents;

По указанным синтаксическим правилам: 012, 4323.3, 3.56 — это примеры незапрещенных представлений цены в евро. Аналогично, 12.,.10 — это примеры недопустимых представлений цены в евро.

D.2 Типы формата значений данных

Грамматика, определенная в данном Приложении, определяет восемь различных типов форматов значения: четыре количественных и пять неколичественных форматов значения.

В следующем разделе мы определяем мета-идентификаторы, используемые для описания указанных форматов. В Разделе D.5 мы определяем синтаксическое правило для четырех мета-идентификаторов, представляющих четыре количественных формата значения, вместе с их смыслом на уровне значений. В Разделе D.6 мы определяем мета-идентификаторы для пяти неколичественных форматов значения, вместе с их смыслом на уровне значений.

D.3 Мета-идентификаторы, используемые для определения форматов

Мета-идентификаторы, используемые в грамматике, определяющей различные форматы значений:

dot = '.'; (точка)

decimalMark = '.'; (десятичная точка) **exponentIndicator** = 'E'; (показатель степени)

numeratorIndicator = 'N'; (показатель числителя)

denominatorIndicator = 'D'; (показатель знаменателя)

leadingDigit = T | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'; (первая цифра)

lengthOfExponent (длина показателя)= **leadingDigit (**первая цифра), **{trailingDigit}** (последующая цифра);

lengthOfIntegerPart (длина целой части)= (**leadingDigit, {trailingDigit}**); **lengthOfNumerator** (длина числителя) = **leadingDigit, {trailingDigit}; lengthOfDenominato**r (длина знаменателя) = **leadingDigit**, **{trailingDigit}**; **lengthOfFractionalPart** (длина дробной части) = (**leadingDigit, {trailingDigit}**) | '0'; l**engthOfIntegralPart** (длина целой части)= (**leadingDigit**, **{trailingDigit})|** '0';

lengthOfNumber (длина числа) = **leadingDigit, {trailingDigit}**;
trailingDigit (последующая цифра) = '0' | T I '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'; **signedExponent** (показатель степени со знаком) = 'S';
signedNumber (число со знаком)= **space** (пробел),'S';
space (пробел) = '';
variableLengthIndicator (индикатор длины переменной) = '..';
decimalMark (десятичный знак): разделитель между целой и дробной частями числа в формате NR2 или NR3.
leadingDigit: первая цифра числа, включающая одну или несколько цифр.
trailingDigit: одна из последующих цифр, образующая число (кроме первой).

Примечание — Если число состоит только из одной цифры, то в нем последующих цифр нет.

D.4 Количественный формат значения

В нижеследующих четырех подразделах определены четыре синтаксических правила для количественного формата значения и их смысл, необходимый для представления значения. Они могут использоваться для свойств с нижеследующими типами данных:

- числовой тип **number_type** или любой его подтип;
- тип уровня **level_type** с типом значения **value_type** либо действительной меры **real_measure_type,** либо целой меры **int_measure_type;**
- списочный тип **list_type,** тип множества **set_type,** упакованный тип **bag_type,** тип массива **array_type,** или множества с подмножеством ограничений **set_with_subset_constraint_type,** для которых тип значения **value_type** — это тип числа **number_type** или любой его подтип.

Примечание 1 — Указанные выше типы **list_type, set_type, bag_type, array_type** и **set_with_subset_constraint_type** определены в ИСО 13584-25.

Примечание 2 — Для неколичественного целого типа **non_quantitative_int_type** формат значения используется для задания кода.

Примечание 3 — Значение данного атрибута должно быть совместимо с типом данных свойства: оно не должно изменять указанный тип данных или должно игнорироваться.

Пример — Формат значения NR2 не совместим с целым типом int_type, так как целые значения не имеют дробной части.

D.4.1 Формат значения NR1

Синтаксис значений NR1 указывает формат целого значения свойства.
Синтаксическое правило:
NRIValue = 'NRI', ((signedNumber, variableLengthIndicator) |
(signedNumber, space) | variableLengthIndicator | space),
lengthOfNumber;
Смысл компонентов формата NR1:
- 'NR1': значение должно быть целым.

Примечание 1 — Значение NR1 не должно содержать пробелов.

- lengthOfNumber: количество цифр числа.

Примечание 2 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр может быть меньше.

- signedNumber: если указано число со знаком signedNumber, то оно должно быть либо положительным, либо отрицательным, либо нулем. Положительное число может иметь знак '+'. Отрицательное число имеет знак '-'. Нуль не может иметь знак '.
- variableLengthIndicator: если задан индикатор длины переменной, то количество цифр соответствующего числа не должно превышать заданной длины, т. е. lengthOfNumber.

D.4.2 Формат значения NR2

Синтаксис значения NR2 указывает формат действительного значения свойства, не требующего указания показателя степени.
Синтаксическое правило:
NR2Value = 'NR2', ((signedNumber, variableLengthIndicator) |
(signedNumber, space) | variableLengthIndicator | space),
lengthOfIntegralPart, decimalMark, lengthOfFractionalPart;
Компоненты значения формата NR2:
- 'NR2': значение должно быть действительным.

Примечание 1 — Численные значения NR2 не должны содержать пробелов.

- lengthOfFractionalPart: число цифр дробной части числа.

Примечание 2 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр дробной части числа может быть меньше.

Примечание 3 — Длина дробной части числа lengthOfFractionalPart неявно указывает рекомендованную точность числа. Фактическая точность числа, из которого выведено значение, может быть выше той, что указана здесь.

- lengthOf IntegralPart: количество цифр целой части числа.

Примечание 4 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр целой части числа может быть меньше.

- signedNumber: если указано число со знаком signedNumber, то оно должно быть либо положительным, либо отрицательным, либо нулем. У положительного числа спереди может быть плюс '+'. У отрицательного числа спереди должен быть минус '–'. Перед нулем минус '–' недопустим.
- variableLengthIndicator: если задан индикатор длины переменной variableLengthIndicator, то либо целая часть, либо дробная часть числа, либо обе эти части должны содержать цифры, количество которых не превышает длину, заданную параметрами lengthOfIntegralPart или lengthOfFractionalPart. Число должно содержать по крайней мере одну цифру.

D.4.3 Формат значения NR3

Синтаксис значения NR3 указывает формат действительного значения свойства, представленного в степенной форме.

Синтаксическое правило:

NR3Value = 'NR3', ((signedNumber, variableLengthIndicator) |
(signedNumber, space) | variableLengthIndicator | space),
lengthOfIntegralPart, decimalMark, lengthOfFractionalPart,
exponentIndicator, [signedExponent], 1engthOfExponent;

Смысл компонент формата значения NR3:

- 'NR3': значение должно быть действительным с основанием степени 10.

Примечание 1 — Число должно содержать, по крайней мере, одну цифру и десятичный знак в мантиссе. Показатель степени имеет, по крайней мере, одну цифру.

Примечание 2 — Значения формата NR3 не имеют пробелов.

- exponentIndicator: разделитель между мантиссой и показателем степени в формате NR3.
- lengthOfExponent: число цифр показателя степени.

Примечание 3 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр показателя степени может быть меньше.

Примечание 4 — Случайные плюсы и минусы или десятичные знаки не считаются параметрами lengthOfNumber, lengthOfIntegralPart, lengthOfFractionalPart или lengthOfExponent.

- lengthOfFractionalPart: количество цифр дробной части мантиссы.

Примечание 5 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр дробной части числа может быть меньше.

Примечание 6 — Параметр длины дробной части числа lengthOfFractionalPart неявно указывает рекомендованную точность числа. Фактическая точность числа, из которого выводится данное значение, может быть выше, чем значение, данное здесь.

- lengthOfIntegralPart: количество цифр целой части мантиссы.

Примечание 7 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр целой части числа может быть меньше.

- signedExponent: если указан показатель степени со знаком signedExponent, то он может быть положительным, отрицательным или нулем. Положительная степень может быть со знаком '+'. Отрицательная степень должна быть со знаком '–'. Нуль не может быть со знаком '–'.
- variableLengthIndicator: если указан индикатор длины переменной variableLengthIndicator, то количество цифр числа (показателя степени) не может превышать значения, указанного в спецификации, т. е. lengthOfIntegralPart, lengthOf FractionalPart или lengthOfExponent. В мантиссе и в показателе степени должна быть, по крайней мере, одна цифра.

D.4.4 Формат значения NR4

Синтаксис значения NR4 указывает формат рационального значения свойства, имеющего целую часть и, возможно, дробную часть со знаменателем и числителем.

Синтаксическое правило:

NR4Value = 'NR4', ((signedNumber, variableLengthIndicator) | (signedNumber,
space) | variableLengthIndicator | space), lengthOfIntegerPart,

numeratorIndicator, lengthOfNumerator, denominatorIndicator, lengthOfDenominator;

Смысл компонент значения формат NR4:

- 'NR4': значение должно быть рациональным числом, представленным либо как целое, либо как дробь, состоящая из числителя и знаменателя, либо как целое и дробь.

Пример — 12 ½ и 12 ¾ — это значения, представленные в формате NR4.

Примечание 1 — В целой части или в числителе и знаменателе должна быть, по крайней мере, одна цифра. Если одна часть дроби содержит цифру, то другая часть дроби также должна содержать цифры. Все три части числа могут содержать цифры.

Примечание 2 — Формат NR4 не допускает пробелов.

- numeratorIndicator: разделитель между целой частью и дробной частью в формате NR4.
- lengthOfNumerator: количество цифр числителя.

Примечание 3 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр числителя может быть меньше.

Примечание 4 — Если значение рационального числа полно представлено своей целой частью, то ни числителя дроби, ни знаменателя может не быть.

- denominatorIndicator: разделитель между числителем и знаменателем дроби в формате NR4.
- lengthOfDenominator: количество цифр знаменателя.

Примечание 5 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр знаменателя может быть меньше.

Примечание 6 — Если значение рационального числа полно представлено его целой частью, то ни числителя дроби, ни знаменателя дроби может не быть.

- lengthOfIntegerPart: количество цифр целой части рационального числа.

Примечание 7 — Если указан индикатор длины переменной variableLengthIndicator, то фактическое количество цифр целой части может быть меньше.

- variableLengthIndicator: если указан индикатор длины переменной variableLengthIndicator, то количество цифр во всех трех частях рационального числа не должно превышать значения, указанного в спецификации, т. е. lengthOfIntegralPart, lengthOfNumerator, или lengthOfDenominator. В целой части числа или в обеих частях дробной части числа должна быть, по крайней мере, одна цифра.

D.5 Неколичественные форматы значений

В нижеследующих пяти подразделах определены пять синтаксических правил для неколичественных форматов значений и рассмотрен их смысл. Они используются для свойств с нижеследующим типом данных:

- строчный тип **string_type** или любой его подтип;
- списочный тип **list_type,** тип множества **set_type,** упакованный тип **bag_type,** тип массива **array_type** или тип множества с подмножеством ограничений **set_with_subset_constraint_type,** для которого тип значения **value_type** — это строчный тип **string_type** или любой его подтип.

Примечание 1 — Типы **list_type, set_type, bag_type, array_type и set_with_subset_constraint_type** определены в ИСО 13584-25.

Примечание 2 — Для переводимого строчного типа **translatable_string_type** рассматриваемый формат значения применяется для представления строк на заданном языке.

Примечание 3 — Для неколичественного кодового типа **non_quantitative_code_type** формат значения применяется к коду.

Неколичественное значение представляется строкой с символами. Длина строки либо устанавливается прямо (путем прямого указания верхнего предела для количества символов), либо путем указания, что общее число символов может быть любым целым кратным некоторой заданной длины.

Синтаксическое правило:

factor = leadingDigit, {trailingDigit};

numberOfCharacters = (leadingDigit, {trailingDigit})|('(nx', factor,')');

Смысл компонент фактора factor:

- factor: если фактор factor указан, то количество символов numberOfCharacters должно быть любым целым кратным значению, указанному в factor. factor не должен содержать значения «нуль».
- numberOfCharacters: определяет максимальное количество символов в строке.

D.5.1 Формат алфавитного значения

«Формат алфавитного значения (A)» определяет формат значения строки, содержащей буквы. Таким образом, содержание строки составляется из символов строки 00, клетки 20, клетки 40 7E, и клетки C0 FF Базовой Многоязыковой Плоскости (БМП) (Плоскость 00 Группы 00), описанной в ИСО/МЭК 10646-1.

Примечание 1 — Из-за потенциальных проблем интеграции содержания значений внутри компонентов одной системы или нескольких систем, используемые символы, по возможности, ограничиваются множеством G0 по ИСО/МЭК 10646-1 и/или строкой 00 и столбцами 002—007 по ИСО/МЭК 10646-1.

Примечание 2 — Для альтернативных языков, используемых для перевода, требуется доступ к некоторым символам или идеограммам особого набора символов (соответствующего языка) в соответствии с стандартом Unicode. В большинстве случаев между символами исходного языка и символами целевого языка имеет место соотношение 1:1.

Пример — Идеограмма CJK (Chinese-Japanese-Korean, т.е. Китай-Япония-Корея).

Синтаксическое правило:

AValue = 'A', (space | variableLengthIndicator), numberOfCharacters;

Смысл компонент A-формата значения:

– 'A': значение должно быть строкой, несколькими подстроками или буквами.

- variableLengthIndicator: если указан индикатор длины переменной variableLengthIndicator, то строка может содержать меньше символов, чем указано параметром numberOfCharacters. В строке должен быть, по крайней мере, один символ.

D.5.2 Формат значения со смешанными символами

«Смешанный формат значения (M)» определяет строку, содержащую любой символ, установленный в разделе D.7.

Примечание — Для альтернативных языков, используемых для перевода, необходим доступ к символам или идеограммам особых наборов символов соответствующего языка в соответствии с стандартом Unicode.

Пример — Идеограмма CJK (Chinese-Japanese-Korean, т.е. Китай-Япония-Корея).

Синтаксическое правило:

MValue = 'M', (space | variableLengthIndicator), numberOfCharacters;

Смысл компонент M-формата значения:

– 'M': значение должно быть строкой или несколькими подстроками.

- variableLengthIndicator: если указан индикатор длины переменной variableLengthIndicator, то строка может содержать меньше символов, чем указано параметром numberOfCharacters. Строка должна содержать, по крайней мере, один символ.

D.5.3 Формат численного значения

«Формат численного значения (N)» определяет строку, содержащую только цифры. Таким образом, необходимое значение составляется из строки 00, клетки 2B, клетки 2D, клеток 30—39 и клетки 45 Базовой Многоязыковой Плоскости (БМП) (Плоскость 00 Группы 00) в соответствии с ИСО/МЭК 10646-1.

Примечание — Для альтернативных языков, используемых для перевода, необходим доступ к символам или идеограммам особого набора символов соответствующего языка в соответствии с стандартом Unicode.

Пример — Таблица D.2 устанавливает соответствие европейских цифр «0» — «9» и арабских цифр.

Таблица D.2 — Соответствие европейских цифр и арабских цифр

Европейские цифры	9	8	7	6	5	4	3	2	1	0
Арабские цифры	٩	٨	٧	٦	٥	٤	٣	٢	١	٠

Синтаксическое правило:

NValue = 'N', (space | variableLengthIndicator), numberOfCharacters;

Смысл компонент значения N-формата:

- 'N': значение должно быть строкой, несколькими подстроками или цифрами.

- variableLengthIndicator: если указан индикатор длины переменной variableLengthIndicator, то строка может содержать меньше символов, чем указано параметром numberOfCharacters. Строка должна содержать, по крайней мере, один символ.

D.5.4 Смешанный алфавитно-цифровой формат

«Смешанный алфавитно-цифровой формат (X)» определяет строку, содержащую буквенно-цифровые символы, т.е. любую комбинацию символов значения A-формата и N-формата.

Примечание — Для альтернативных языков, используемых для перевода, необходим доступ к символам и идеограммам особого набора символов соответствующего языка в соответствии с стандартом Unicode.

Синтаксическое правило:

XValue = 'X', (space | variableLengthIndicator), numberOfCharacters;

Смысл компонент значения X-формата:

– 'X': значение должно быть строкой, несколькими подстроками, или буквенно-цифровым выражением, т. е. любой комбинацией букв и цифр;

- variableLengthIndicator: если указан индикатор длины переменной variableLengthIndicator, то строка может содержать меньше символов, чем указано параметром numberOfCharacters. Строка должна содержать, по крайней мере, один символ.

D.5.5 Формат бинарного значения

«Формат бинарного значения (B)» определяет значение строки, содержащей бинарные символы, т. е. «0» или «1». Таким образом, строка составляется из символов строки 00, клеток 30 или 31 Базовой Многоязыковой Плоскости (БМП) (Плоскость 00 Группы 00) в соответствии с ИСО/МЭК 10646-1.

Примечание — Для альтернативных языков, используемых для перевода, необходим доступ к символам и идеограммам особого набора символов соответствующего языка в соответствии с стандартом Unicode.

Синтаксическое правило:

BValue = 'B', (space | variableLengthIndicator), numberOfCharacters;

Смысл компонент B-формата значения:

– 'B': значение должно быть строкой или несколькими подстроками, содержащими бинарные значения, т. е. символы «0» или «1» или их последовательности.

- variableLengthIndicator: если указан индикатор длины переменной variableLengthIndicator, то строка может содержать меньше символов, чем указано параметром numberOfCharacters. Строка должна содержать, по крайней мере, один символ.

D.6 Пример задания значения

Таблица D.3 ниже содержит некоторые примеры задания значений, выраженных числами и строками в соответствии с указанными выше форматами.

Таблица D.3 — Пример задания значения

Формат значения	Пример задания значения
NR1 3	123; 001; 000;
NR1..3	123; 87; 5
NR1S3	+123; +000;
NR1S..3	– 123; +1; 0; – 12;
NR2 3.3	123.300; 000.400; 000.420;
NR2..3.3	321.233; 1.234; 23.56; 9.783;.72; 324.
NR2S 3.3	– 123.123; +123.300;
NR2S..3.3	– 123.123; +12.3; 0.1; +.4; – 3.; 0.;.0;
NR3 3.3E4	123.123E0004; 003.000E1000;
NR3 3.3ES4	123.123E+0004; 123.123E0004; 123.000E-0001
NR3..3.3E4	123.123E0004;.123E0001; 5.E1234
NR3S 3.3ES4	+ 123.123E+0004; 123.000E-0001;
NR3S..3.3ES4	– 123.123E+0004; +1.00E-01;.0E0; +3. E-1; – .2E-1000;
NR4 3N2D2	001 02/03; 012 00/01; 123 03/04; 000 01/04
NR4..3N2D2	$1\ ^1/_2$; 12; $123\ ^3/_4$; %;
NR4S 3N2D2	+001 02/03; – 012 00/01; 123 03/04; – 000 01/04
NR4S..3N2D2	$-\ 1\ ^1/_2$; 12; +123 %; %;
A19	My name is Reinhard, abcdefghijklmnopqrs
A..3	Abc; de; G
X..5	A23RN1; B1; ca
M..10	A23RN1; B1; ca. 256 U-m;

Окончание таблицы D.3

Формат значения	Пример задания значения
N (nx5)	12345; 1234512345; 222223333344444;
N..(nx5)	1234; 12345; 34512345; 1234512345; 23333344444; 222223333344444; – 3; 5E2;
B 1	0; 1;
B3	011; 101;

D.7 Символы по ИСО/МЭК 10646-1

В смешанном формате значений (M) используются нижеследующие символы (см. D.5.2):

- все символы из строки 00 Базовой Многоязыковой Плоскости (БМП) (Плоскости 00 Группы 00) в соответствии с ИСО/МЭК 10646-1;
- символы из других строк Базовой Многоязыковой Плоскости в соответствии с ИСО/МЭК 10646-1 (см. таблицу D.4).

Для альтернативных языков, используемых для перевода, необходим доступ к полному набору символов, определенному стандартом Unicode.

Примечание 1 — Из-за потенциальных проблем с интерпретацией значений внутри компонентов одной системы или нескольких систем, рекомендуется, по возможности, брать символы только из набора G0, определенного ИСО/МЭК 10646-1, и/или из строки 00 и столбцов 002—007 ИСО/МЭК 10646-1.

Примечание 2 — Международный Стандарт Unicode опубликован Консорциумом Unicode. Адрес: P. O. Box 391476, Mountain View, CA 94039-1476, U. S. A., gost.e-ecolog.ru

Таблица D.4 — Символы из других рядов Основной Многоязыковой Плоскости по ИСО/МЭК 10646-1

Символ	Имя	Ряд	Клетка
	CARON («птичка» над буквой как у «Й»)	02	C7
≡	IDENTICAL TO (идентичность множеств)	22	61
∧	LOGICAL AND (логическая операция «И»)	22	27
∨	LOGICAL OR (логическая операция «ИЛИ»)	22	28
∩	INTERSECTION (пересечение множеств)	22	29
∪	UNION (операция «включающее ИЛИ»)	22	2A
⊂	SUBSET OF (включать)	22	82
⊃	SUBSET OF (быть включенным)	22	83
⇐	LEFTWARDS DOUBLE ARROW (из второго следует первое)	21	D0
⇒	RIGHTWARDS DOUBLE ARROW (из первого следует второе)	21	D2
∴	THEREFORE (следовательно)	22	34
ϴ	BECAUSE (потому что)	22	35
ϵ	ELEMENT OF (является элементом чего-либо)	22	08
э	CONTAINS AS MEMBER (включать как элемент)	22	0B
⊆	SUBSET OR EQUAL TO (само множество включается в другое множество как подкласс)	22	86
⊇	SUPERSET OR EQUAL TO (множество включает другое множество как подкласс)	22	87
∫	INTEGRAL (интеграл)	22	2B
∮	CONTOUR INTEGRAL (интеграл по замкнутому контуру)	22	2E

Продолжение таблицы D.4

Символ	Имя	Ряд	Клетка
∞	INFINITY (бесконечность)	22	1E
∇	NABLA (дифференциальный набла-оператор)	22	07
∂	PARTIAL DIFFERENTIAL (частная производная)	22	02
~	TILDE OPERATOR (тильда-оператор: разность между)	22	3C
≈	ALMOST EQUAL TO (приближенное равенство)	22	48
≃	ASYMPTOTICALLY EQUAL TO (асимптотическое равенство)	22	43
≅	APPROXIMATELY EQUAL TO (подобие)	22	45
≤	LESS THEN OT EQUAL TO (меньше или равно)	22	64
≠	NOT EQUAL TO (не равно)	22	60
≥	GREATER TAN OR EQUAL TO (больше или равно)	22	65
⇔	LEFT RIGHT DOUBLE ARROW (если и только если)	21	D4
¬	NOT SIGN (знак «не»: не больше, не меньше)	00	AC
∀	FOR ALL (всех)	22	00
∃	THERE EXISTS (существует)	22	03
ℵ	HEBREW LETTER ALEF (буква «алеф» древнееврейского языка)	05	D0
□	WHITE SQUARE (оператор Д'Аламбера)	25	A1
‖	PARALLEL (параллельность)	22	25
Γ	GREEK CAPITAL LETTER GAMMA (греческая большая буква «гамма»)	03	93
Δ	GREEK CAPITAL LETTER DELTA (греческая большая буква «дельта»)	03	94
⊥	UPTACK (ортогональность)	22	A5
∠	ANGLE (угол)	22	20
⊾	RIGHT ANGLE WITH ARC (правый угол с дугой)	22	BE
Θ	GREEK CAPITAL LETTER THETA (греческая заглавная буква «тета»)	03	98
⟨	LEFT POINTING ANGLE BRACKET(левая угловая скобка)	23	29
⟩	RIGHT POINTING ANLE BRACKET (правая угловая скобка)	23	2A
Λ	GREEK CAPITAL LETTER LAMBDA (греческая заглавная буква «лямбда»)	03	9B
′	PRIME (знак штриха «прим»)	20	32
″	DOUBLE PRIME (знак двойного штриха «прим»)	20	33
Ξ	GREEK CAPITAL LETTER XI (греческая заглавная буква «кси»)	03	9E
∓	MINUS-OR-PLUS SIGN (знак минус-плюс)	22	13
Π	GREEK CAPITAL LETTER PI (греческая заглавная буква «пи»)	03	A0

Продолжение таблицы D.4

Символ	Имя	Ряд	Клетка
²	SUPERSCRIPT TWO (верхний индекс 2)	00	B2
Σ	GREEK CAPITAL LETTER SIGMA (греческая заглавная буква «сигма»)	03	A3
×	MULTIPLICATION SIGN (умножение)	00	D7
³	SUPERSCRIPT THREE (верхний индекс 3)	00	B3
ϒ	GREEK CAPITAL LETTER UPSILON (греческая заглавная буква «ипсилон»)	03	A5
Φ	GREEK CAPITAL LETTER PHI (греческая заглавная буква «фи»)	03	A6
·	MIDDLE DOT (точка-разделитель)	00	B7
Ψ	GREEK CAPITAL LETTER PSI (греческая заглавная буква «пси»)	03	A8
Ω	GREEK CAPITAL LETTER OMEGA (греческая заглавная буква «омега»)	03	A9
∅	EMPTY SET (пустое множество)	22	05
⇀	RIGHTWARDS HARPOON WITH BARB UPWARDS («правый гарпун с зазубриной сверху» — черта над вектором)	21	C0
√	SQUARE ROOT (квадратный корень — радикал)	22	1A
f	LATIN SMALL LETTER F WITH HOOK (латинская строчная буква «f с крючком» — обозначение функции)	01	92
∝	PROPORTIONAL (пропорциональность)	22	1D
±	PLUS-MINUS SIGN (знак плюс-минус)	00	B1
°	DEGREE SIGN (градус)	00	B0
α	GREEK SMALL LETTER ALPHA (греческая строчная буква «альфа»)	03	B1
β	GREEK SMALL LETTER BETA (греческая строчная буква «бета»)	03	B2
γ	GREEK SMALL LETTER GAMMA (греческая строчная буква «гамма»)	03	B3
δ	GREEK SMALL LETTER DELTA (греческая строчная буква «дельта»)	03	B4
ε	GREEK SMALL LETTER EPSILON (греческая строчная буква «эпсилон»)	03	B5
ζ	GREEK SMALL LETTER ZETA (греческая строчная буква «дзета»)	03	B6
η	GREEK SMALL LETTER ETA (греческая строчная буква «эта»)	03	B7
θ	GREEK SMALL LETTER THETA (греческая строчная буква «тета»)	03	B8
ι	GREEK SMALL LETTER IOTA (греческая строчная буква «иота»)	03	B9
κ	GREEK SMALL LETTER KAPPA (греческая строчная буква «каппа»)	03	BA

Окончание таблицы D.4

Символ	Имя	Ряд	Клетка
λ	GREEK SMALL LETTER LAMBDA (греческая строчная буква «лямбда»)	03	BB
μ	GREEK SMALL LETTER MU (греческая строчная буква «мю»)	03	BC
ν	GREEK SMALL LETTER NU (греческая строчная буква «ню»)	03	BD
ξ	GREEK SMALL LETTER XI (греческая строчная буква «кси»)	03	BE
‰	PER MILLE SIGN (на тысячу, промилле)	20	30
π	GREEK SMALL LETTER PI (греческая строчная буква «пи»)	03	C0
ρ	GREEK SMALL LETTER RHO (греческая строчная буква «ро»)	03	C1
σ	GREEK SMALL LETTER SIGMA (греческая строчная буква «сигма»)	03	C3
÷	DIVISION SIGN (деление)	03	F7
τ	GREEK SMALL LETTER TAU (греческая строчная буква «тау»)	03	C4
υ	GREEK SMALL LETTER UPSILON (греческая строчная буква «ипсилон»)	03	C5
φ	GREEK SMALL LETTER PHI (греческая строчная буква «фи»)	03	C6
χ	GREEK SMALL LETTER CHI (греческая строчная буква «хи»)	03	C7
ψ	GREEK SMALL LETTER PSI(греческая строчная буква «пси»)	03	C8
ω	GREEK SMALL LETTER OMEGA (греческая строчная буква «омега»)	03	C9
†	DAGGER (знак ссылки)	20	20
←	LEFTWARDS ARROW (стрелка влево)	21	90
↑	UPWARDS ARROW (стрелка вверх)	21	91
→	RIGHTWARDS ARROW (стрелка вправо)	21	92
↓	DOWNWARDS ARROW (стрелка вниз)	21	93
‾	OVERLINE (черта над символом)	20	3E

Приложение Е
(обязательное)

Регистрация информационного объекта

Е.1 Идентификация документа

Для однозначной идентификации информационного объекта в открытых системах, настоящей части ИСО 13584 назначен нижеследующий идентификатор объекта:

{iso standard 13584 part (42) version(3)}

Его значение определено в ИСО/МЭК 8824-1. Описание дано в ИСО 10303-1.

Е.2 Идентификация схемы

Е.2.1 Словарная схема ISO13584_IEC61360_dictionary_SCHEMA

Для однозначной идентификации словарной схемы **ISO13584_IEC61360_dictionary_SCHEMA** в открытой информационной системе, ей назначен нижеследующий идентификатор объекта:

{iso standard 13584 part (42) version(3) schema(1)
ISO13584-IEC61360-dictionary-schema(1)}

(см. раздел F.3). Смысл данного значения определен в ИСО/МЭК 8824-1, описание дано в ИСО 10303-1.

Е.2.2 Схема языкового ресурса ISO13584_IEC61360_language_resource_SCHEMA

Схеме **ISO13584_IEC61360_language_resource_SCHEMA** назначен нижеследующий идентификатор объекта:

{iso standard 13584 part (42) version(3) schema (1)
ISO13584-IEC61360-language-resource-schema(2)}

(см. раздел F.4). Смысл данного значения определен в ИСО/МЭК 8824-1, описание дано в ИСО 10303-1.

Е.2.3 Схема ограничения класса ISO13584_IEC61360_class_constraint_SCHEMA

Схеме **ISO13584_IEC61360_class_constraint_SCHEMA** назначен нижеследующий идентификатор объекта:

{iso standard 13584 part (42) version(1) schema (1)
ISO13584-IEC61360-class-constraint-schema(3)}

(см. раздел F.5). Смысл данного значения определен в ИСО/МЭК 8824-1, описание дано в ИСО 10303-1.

Е.2.4 Условная схема класса предметов ISO13584_IEC61360_item_class_case_of_SCHEMA

Схеме класса предметов **ISO13584_IEC61360_item_class_case_of_SCHEMA** назначен нижеследующий идентификатор объекта:

{iso standard 13584 part (42) version(2) schema (1)
ISO13584-IEC61360-item-class-case-of-schema (4)}

(см. раздел F.6). Смысл данного значения определен в ИСО/МЭК 8824-1, описание дано в ИСО 10303-1.

Приложение F
(справочное)

Подмножество общей словарной схемы МЭК/ИСО, задокументированное в настоящей части ИСО 13584

Данное Приложение рассматривает общее подмножество общего словаря ИСО/МЭК, задокументированное в ИСО 13584-25 и МЭК 61360-5. Нормативная версия данного подмножества опубликована в МЭК 61360-2. Настоящее справочное приложение дублирует нормативное содержание МЭК 61360-2 и дает (с помощью примечаний) некоторые дополнительные пояснения к настоящей части ИСО 13584.

F.1 Общие положения

F.1.1 Область и цель применения общей словарной модели ISO13584/IEC61360

Область применения общего словаря ИСО/МЭК, основанного на использовании рассматриваемой схемы, получается в результате пересечения областей применения двух базовых стандартов:

- МЭК 61360-1, разработанный МЭК/ТК 3/ПК 3D;
- настоящая часть ИСО 13584, разработанная ИСО/ТК 184/ПК4/РГ2.

Рассматриваемая модель языка EXPRESS позволяет получить общую формальную модель двух документов и способствует их гармонизации.

Соответствующие части разделов (с описанием областей применения) даны ниже.

Цитата из МЭК 61360-1: «Настоящая часть МЭК 61360 дает прочную базовую основу для ясного и однозначного определения характеристических свойств (типов элементов данных) всех элементов электротехнических систем: от базовых компонентов — до подсборок и полных систем. Несмотря на то что принципы и методы данного стандарта уже рассматривались в контексте, составляющем базис обмена информацией об электрических/электронных компонентах, эти принципы и методы могут быть также использованы в областях, лежащих вне исходной концепции, например, при создании сборок компонентов, электротехнических систем и подсистем».

Цитата из области применения настоящей части ИСО 13584: «Настоящая часть ИСО 13584 дает описания ...

- атрибутов, предоставляемых поставщиками информации для описания классов характеризации и свойств деталей;
- (...)
- спецификаций атрибутов информационной модели языка EXPRESS, обеспечивающих обмен указанными словарными данными».

Целью данной схемы языка EXPRESS является создание формальной модели данных, соответствующей областям применения (смотри выше) и, таким образом, получение компьютерного представления указанных данных и обеспечение их обмена.

Целью также является создание общей информационной модели для работы обоих технических комитетов. Таким образом, обеспечивается практическая реализация рассматриваемых словарных систем, содержащих данные, полученные в соответствии со стандартами, разработанными указанными комитетами.

F.1.2 Взаимодействие ИСО 13584 и МЭК 61360

Подмножество общей словарной модели ИСО 13584/МЭК 61360, задокументированной в настоящей части ИСО 13584, может быть использовано, как в контексте ИСО 13584, так и в контексте МЭК 61360 для обмена простых словарных данных.

Для словарей, требующих расширенных возможностей (например, внешних документов, ассоциированных со словарными данными, свойства значений которых имеют совокупную структуру или содержат описания функциональных моделей), расширения схематик, задокументированных в настоящей части ИСО 13584, определены совместно ИСО/ТК 184/ПК 4/РГ 2 и МЭК/ТК 3/ПК 3D. Схематики, задокументированные в настоящей части ИСО 13584, и указанные расширения представляют собой новую общую словарную модель ИСО 13584/МЭК 61360, используемую как в контексте ИСО 13584, так и в контексте МЭК 61360. Готовая общая словарная модель ИСО 13584/МЭК 61360 задокументирована как в ИСО 13584-25, так и в МЭК 61360-5. Определение в среде расширенного языка разметки XML задокументировано в ИСО 13584-32.

Область применения серии ИСО 13584 шире, чем область применения серии стандартов МЭК 61360: последний адресован только онтологиям продуктов, тогда как первый адресован как онтологиям продуктов, так и библиотекам. И на самом деле, МЭК 13584-25 обеспечивает ряд дополнительных опций, поддерживаемых при практической реализации. Указанные опции сгруппированы в классы соответствия. Соответствие конкретному классу соответствия требует, чтобы поддерживались все сущности, типы и ассоциированные ограничения, определенные как части данного класса. Поддержка конкретного класса соответствия требует поддержки всех опций, указанных в данном классе.

Классы соответствия от №1 до №4 ИСО 13584-25 точно соответствуют классам соответствия от №1 до №4 МЭК 61360-5. Они предназначены для обмена словарей более сложных, чем те, что могут обмениваться только с помощью подмножества, задокументированного как в настоящей части ИСО 13584, так и в МЭК 61360-2.

Оба технических комитета договорились не вносить изменения и/или не модифицировать представленную модель языка EXPRESS независимо друг от друга, чтобы гарантировать гармонизацию и повторное использование результатов работы обоих комитетов. Таким образом, запросы на изменения стандартов следует направлять обоим комитетам сразу. Указанные запросы рассматриваются обоими комитетами, а затем производится модификация информационной модели языка EXPRESS.

F.2 Основная структура общей словарной схемы

Данный подраздел содержит объяснение основных конструктивов ресурса, используемых общей словарной схемой:

- **dictionary_element** — любой элемент, определенный в словаре;
- **supplier_element** — содержит данные о поставщиках словарных элементов (классы, свойства, типы данных); класс, моделирующий словарные элементы классов (семейств), описанные свойствами;
- **property_DET** — словарный элемент свойства;
- **data_type** — определяет тип свойства.

Указанные части словарной схемы детально представлены **ISO13584_IEC61360_dictionary_schema** в разделе 5.

В рассматриваемом представлении общей словарной схемы некоторые обзорные диаграммы даны как модели планирования (смотри рисунки 1—11). Указанные модели планирования используют графические обозначения EXPRESS-G для языка EXPRESS. Для упрощения рассматриваемых диаграмм некоторые соотношения, определенные в модели EXPRESS, опущены. На рисунке 1 приведена модель планирования основной структуры общей словарной схемы. Большинство указанных рисунков содержат обзорные модели (модели планирования). Их уровень детализации соответствует уровню рассмотрения.

Для упрощения рассматриваемых диаграмм некоторые соотношения, определенные в модели EXPRESS, опущены. На рисунке F.1 ниже дана модель планирования основной структуры общей словарной модели ИСО 13584/МЭК 61360.

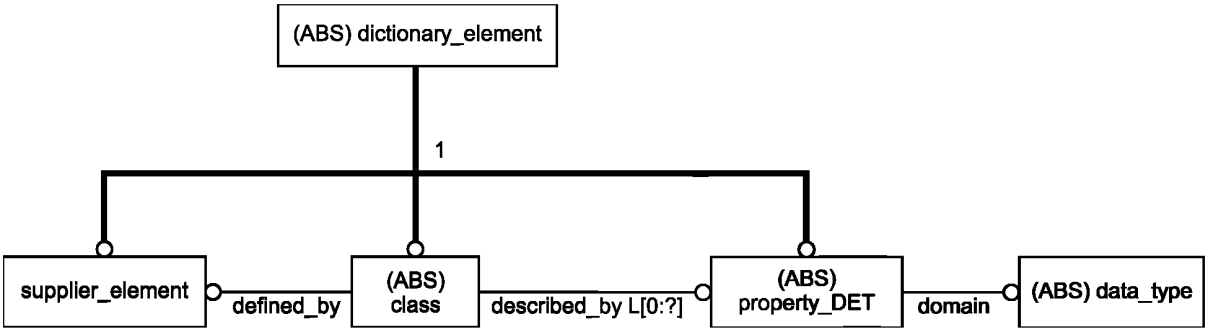


Рисунок F.1 — Словарная схема

(ABS) dictionary_element	Словарный элемент
supplier_element	Элемент поставщика
defined_by	Определен с помощью ...
(ABS) class	Класс
described_by L[0:?]	Описан массивом ...
(ABS property_DET)	Тип элемента данных свойства
domain	Область
(ABS) data_type	Тип данных

Большинство из этих цифр содержат обзор модели (или моделей планирования), но показывают только уровень детализации, который подходит в определенном месте.

F.3 Стандартная словарная схема ISO13584_IEC61360_dictionary _schema

Данный раздел, представляющий основную часть общей информационной модели ИСО 13584-42 и серии стандартов МЭК 61360, содержит полный листинг (на языке EXPRESS) словарной схемы, аннотированный с комментариями и пояснительным текстом. Порядок расположения текста в данном разделе определен, главным образом, порядком, установленным языком EXPRESS. Важность этого — вторична.

F.3.1 Введение в схему

Этот подраздел устанавливает описание схемы на языке EXPRESS с помощью объявлений переменных и констант, а также внешних ссылок.

F.3.1.1 Словарная схема

Прежде всего схему нужно объявить.

Пример представления на языке EXPRESS:

```
*)
SCHEMA ISO13584_IEC61360_dictionary_schema;
(*
```

F.3.1.2 Ссылки на другие схематики

Данный подраздел содержит ссылки на другие схематики EXPRESS, используемые в данной словарной схеме. Их источник указан в соответствующем комментарии.

Пример представления на языке EXPRESS:

```
*)
        REFERENCE FROM support_resource_schema (identifier, label, text);
        REFERENCE FROM person_organization_schema (organization, address);
        REFERENCE FROM measure_schema;
        REFERENCE FROM ISO13584_IEC61360_language_resource_schema;
        REFERENCE FROM ISO13584_IEC61360_class_constraint_schema;
        REFERENCE FROM ISO13584_IEC61360_item_class_case_of_schema;
        REFERENCE FROM ISO13584_external_file_schema
            (external_item,
            external_file_protocol,
            external_content,
            not_translatable_external_content,
            not_translated_external_content,
            translated_external_content,
            language_specific_content,
            http_file,
            http_class_directory,
            http_protocol);
    (*
```

Примечание — Схематика, указанная выше, описана в нижеследующих документах:

support_resource_schema	ИСО 10303
person_organization_schema	ИСО 10303
measure_schema	ИСО 10303
ISO13584_IEC61360_language_resource_schema	МЭК 61360-2
(схема дублирована для удобства в данном документе)	
ISO13584_IEC61360_class_constraint_schema	МЭК 61360-2
(схема дублирована для удобства в данном документе)	
ISO13584_IEC61360_item_class_case_of_schema	МЭК 61360-2
(схема дублирована для удобства в данном документе)	
ISO13584_external_file_schema	ИСО 13584-24

F.3.2 Определения констант

Данный подраздел содержит определения целочисленных констант, используемые позже в определениях типа (см. F3.9).

Пример представления на языке EXPRESS:

```
*)
    CONSTANT
        dictionary_code_len: INTEGER := 131;
        property_code_len: INTEGER := 35;
        class_code_len: INTEGER := 35;
        data_type_code_len:INTEGER := 35;
        supplier_code_len: INTEGER := 149;
        version_len: INTEGER := 10;
        revision_len: INTEGER := 3;
        value_code_len: INTEGER := 35;
        pref_name_len: INTEGER := 255;
        short_name_len: INTEGER := 30;
```

```
        syn_name_len: INTEGER := pref_name_len;
        DET_classification_len: INTEGER := 3;
        source_doc_len: INTEGER := 255;
        value_format_len: INTEGER := 80;
        sep_cv: STRING := '#';
        sep_id: STRING := '#';
    END_CONSTANT;
    (*
```

F.3.3 Идентификация словаря

Сущность «идентификация словаря **dictionary_identification»** позволяет недвусмысленно идентифицировать конкретную версию конкретного словаря конкретного поставщика информации (стандартную или произвольную). Данная сущность содержит код **code,** определенный поставщиком словаря и идентифицирующий словарь, № версии и № пересмотра, характеризующие конкретное состояние данного словаря.

Примечание 1 — Случаи, если № версии и № пересмотра словаря получают приращения, определены в МЭК 61360-1.

Пример представления на языке EXPRESS:

```
    *)
    ENTITY dictionary_identification;
        code: dictionary_code_type;
        version: version_type;
        revision: revision_type;
        defined_by: supplier_bsu;
    DERIVE
        absolute_id: identifier :=
        defined_by.absolute_id + sep_id + code + sep_cv + version;
    UNIQUE
        UR1: absolute_id;
    END_ENTITY; -- dictionary_identification
    (*
```

Определения атрибутов:

code: код, характеризующий словарь.

version: № версии, характеризующий рассматриваемую версию словаря.

revision: № пересмотра, характеризующий пересмотр словаря.

defined_by: поставщик, определяющий словарь.

absolute_id: уникальная идентификация словаря.

Пояснения к тексту программы:

UR1: идентификатор словаря, определенный атрибутом **absolute_id,** является уникальным.

Дополнительные пояснения:

IP1: если словарь определен стандартным документом, содержащим только один словарь, то код словаря должен быть стандартным № документа, описывающего данный словарь, если данный документ определяет только один словарь. Это есть название надлежащего словаря в документе, описывающем его, если данный документ определяет несколько словарей. Если не оговорено противное, то № версии равен 1, и № пересмотра равен 0 для словарей, определенных стандартными документами.

Примечание 2 — Порядок представления стандартных №№ стандартных документов установлен в разделах 5.1 и 5.2 ИСО 13584-26:2000.

F.3.4 Базовые семантические единицы: определения и использование словаря

F.3.4.1 Требования обмена

При обмене данными словарей и библиотек деталей данные обычно делят на части. Например, словарь можно обновить несколькими классами, которые устанавливают их суперкласс путем ссылки на уже существующие классы, или (если обменивается содержание библиотек) путем ссылки на словарные элементы (только ссылки без включения). Ссылка на словарные данные должна быть недвусмысленной и непротиворечивой.

Таким образом, первое требование — это иметь возможность обменивать блоки данных, и второе — иметь соотношения между указанными блоками (см. рисунок F.2).

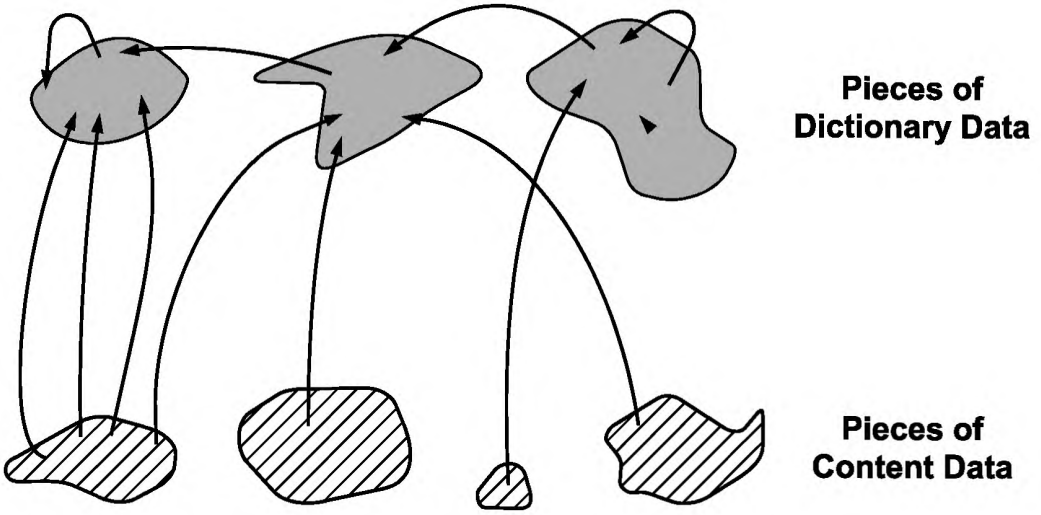


Рисунок F.2 — Блоки данных и соотношения между блоками

Pieces of Dictionary Data	Блоки данных словаря
Pieces of Content Data	Блоки данных содержания

Каждый из указанных блоков соответствует физическому файлу (в соответствии с ИСО 10303-21). Атрибуты языка EXPRESS (см. ИСО 10303-11) могут содержать только ссылки на данные внутри того же физического файла. Таким образом, невозможно использовать атрибуты языка EXPRESS прямо для внутренних ссылок.

F.3.4.2 Трехуровневая архитектура словарных данных

В данном разделе понятие базовой семантической единицы **basic_semantic_unit** (БСЕ) вводится как средство применения указанных внутренних ссылок. БСЕ обеспечивает универсально уникальную идентификацию словарных описаний (см. рисунок 3).

Предположим, что для некоторого контента необходима ссылка на заданное словарное описание.

Пример 1 — Для передачи значения свойства компонента.

Это достигается путем ссылки на базовую семантическую единицу с помощью атрибута словарного определения **dictionary_definition**.

Словарное описание (**dictionary_element**) ссылается на базовую семантическую единицу с помощью атрибута идентификации **identified_by**. Рассматриваемое косвенное соотношение следует из соответствия абсолютных идентификаторов базовой семантической единицы.

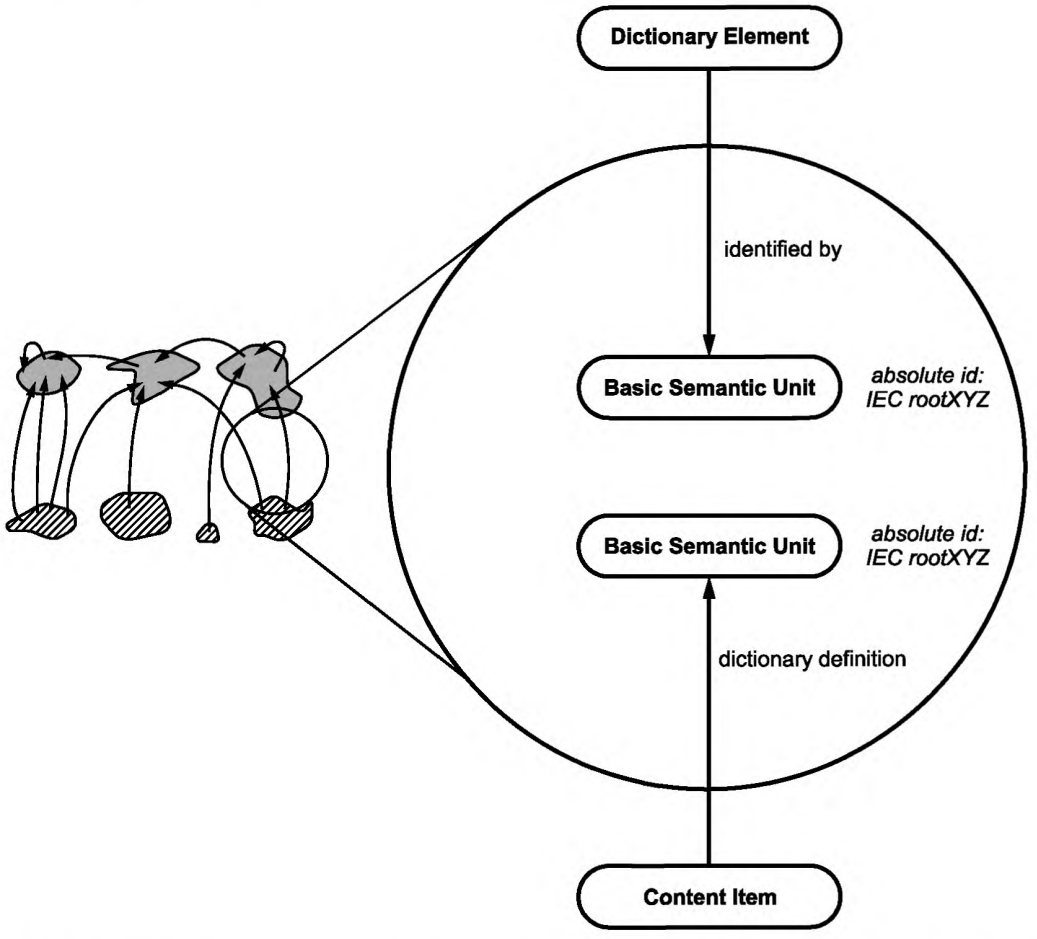


Рисунок F.3 — Практическая реализация «внутреннего» соотношения с помощью базовой семантической единицы

dictionary element	Словарный элемент
identified by	Идентифицирован …
basic semantic unit	Базовая семантическая единица
absolute id:	Абсолютный идентификатор
dictionary definition	Словарное определение
content item	Предмет контента

Отметим, что:

- и словарный элемент, и предмет содержания могут содержаться в одном физическом файле, но не обязательно;

- словарный элемент не всегда нужен для обмена предмета содержания, который на него ссылается. В данном случае предполагается, что словарный элемент уже присутствует в словаре целевой системы. И наоборот, словарные данные можно обменивать без данных контента;

- базовая семантическая единица может быть одной-единственной реализацией в случае, если реализации и словарного элемента, и предмета содержания содержатся в одном физическом файле;

- тот же механизм применяется для ссылок между различными словарными элементами.

Пример 2 — Между классом компонентов и ассоциированными свойствами типов элементов данных property_DET.

БСЕ обеспечивает ссылку на словарное описание в любом необходимом месте.

Пример 3 — Доставка словаря, доставка обновления, доставка библиотеки, обмен данных компонентов.

Данные, ассоциированные со свойством, можно обменивать как пару (**property_BSU,** <значение>). На рисунке F.3 приведена практическая реализация указанного общего механизма.

F.3.4.2.1 Базовая семантическая единица (basic_semantic_unit)

Basic_semantic_unit — это уникальная идентификация словарного элемента **dictionary_element**. БСЕ — аббревиатура базовой семантической единицы.

Пример представления на языке EXPRESS:

```
*)
ENTITY basic_semantic_unit
ABSTRACT SUPERTYPE OF(ONEOF(
        supplier_BSU,
        class_BSU,
        property_BSU,
        data_type_BSU,
        supplier_related_BSU,
        class_related_BSU));
        code: code_type;
        version: version_type;
DERIVE
        dic_identifier: identifier := code + sep_cv + version;
INVERSE
        definition: SET [0:1] OF dictionary_element
                FOR identified_by;
        referenced_by: SET [0:1] OF content_item
                FOR dictionary_definition;
END_ENTITY; -- basic_semantic_unit
(*
```

Определения атрибутов:

code: код, идентифицирующий некоторый словарный элемент.

version: № версии словарного элемента.

dic_identifier: полная идентификация, состоящая из последовательного включения кода и № версии.

definition: ссылка на словарный элемент, идентифицированный указанной БСЕ. Если определение в некотором контексте обмена отсутствует, то предполагается его присутствие в словаре целевой системы.

referenced_by: предметы, использующие словарный элемент, ассоциированный с указанным БСЕ.

F.3.4.2.2 Словарный элемент (Dictionary_element)

Dictionary_element — это полное определение данных, отнесенных (в семантическом словаре) к некоторым понятиям. Для каждого понятия используется свой подтип. Словарный элемент **dictionary_element** ассоциирован с базовой семантической единицей **basic_semantic_unit** (БСЕ), которая уникально идентифицирует данное определение в словаре.

Путем включения атрибута версии в сущность **basic_semantic_unit** получается частичная идентификация словарного элемента (сравни атрибуты № пересмотра **revision** и отметки времени **time_stamps**).

Пример представления на языке EXPRESS:

```
*)
ENTITY dictionary_element
ABSTRACT SUPERTYPE OF(ONEOF(
        supplier_element,
        class_and_property_elements,
        data_type_element));
        identified_by: basic_semantic_unit;
        time_stamps: OPTIONAL dates;
        revision: revision_type;
        administration: OPTIONAL administrative_data;
        is_deprecated: OPTIONAL BOOLEAN;
        is_deprecated_interpretation: OPTIONAL note_type;
WHERE
        WR1: NOT EXISTS (SELF.is_deprecated)
                OR EXISTS (SELF.is_deprecated_interpretation);
END_ENTITY; -- dictionary_element
(*
```

Определения атрибутов:

identified_by: БСЕ, идентифицирующая словарный элемент.

time_stamps: вспомогательные даты создания и обновления словарного элемента.

revision: № пересмотра словарного элемента.

Примечание 1 — Тип атрибута идентификации **identified_by** может быть повторно определен позже для базовой семантической единицы свойства **property_BSU** и базовой семантической единицы класса **class_BSU**. Тогда он может быть использован для кодирования вместе с атрибутом кода БСЕ (для свойства и класса соответственно). Он также может быть использован для кодирования атрибута «№ версии» (для свойства и класса соответственно).

Примечание 2 — Атрибут отметок времени **time_stamps** может быть использован как отправная точка для кодирования (в сущности **dates**) атрибутов свойства и класса «Дата исходного определения», «Дата текущей версии» и «Дата текущего пересмотра» (см. 7.2, 8.2 и F.3.9.2).

Примечание 3 — Атрибут пересмотра **revision** используется для кодирования атрибута свойства и класса «№ пересмотра» (см. 7.2 и 8.2).

administration: вспомогательная информация о жизненном цикле словарного элемента **dictionary_element.**

Примечание 4 — Атрибут **administration** используется для представления информации, относящейся к управлению конфигурацией и к истории трансляции.

is_deprecated: вспомогательная булевская переменная. Если она равна **true**, то данный словарный элемент **dictionary_element** уже нельзя использовать.

is_deprecated_interpretation: обоснование отказа от использования, а также указание порядка интерпретации значений реализаций больше не используемого элемента и его соответствующей **БСЕ**.

Пояснения к тексту программы:

WR1: если существует элемент **is_deprecated,** то существует и элемент **is_deprecated_interpretation**.

Дополнительные пояснения:

IP1: значения реализаций элемента **is_deprecated_interpretation** должны быть определены в момент, если принимается решение об отказе от использования.

На рисунке F.4 представлена модель планирования соотношения между базовой семантической единицей и словарным элементом.

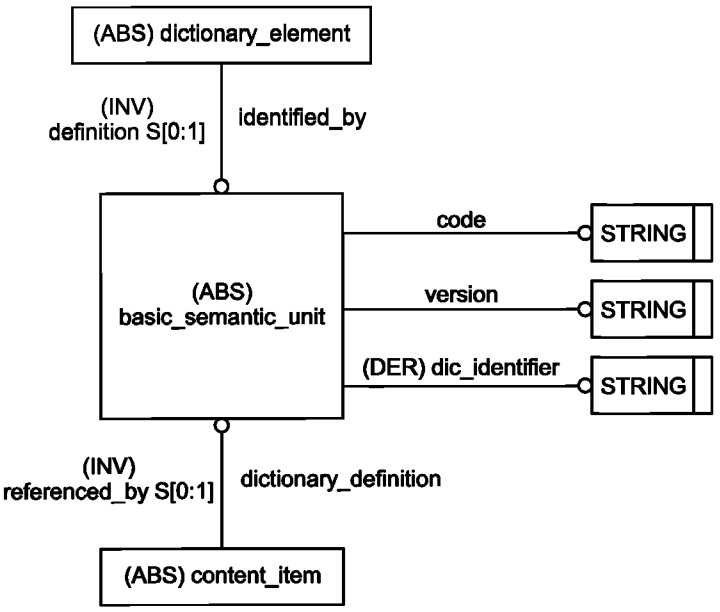


Рисунок F.4 — Соотношение между базовой семантической единицей и словарным элементом

(ABS) dictionary_element	Словарный элемент
(INV) definition S[0:1]	Определение …
identified_by	Идентифицирован …
Code	Код
(ABS) basic_semantic_unit	Базовая семантическая единица
Version	№ версии
String	Строка
(INV) referenced_by S[0:1]	Запрашивается ….

dictionary_definition	Определение словаря
(DER) dic_identifier	Идентификатор словаря
(ABS) content_item	Предмет контента

F.3.4.2.3 Предмет контента (Content_item)

Content_item — это блок данных, ссылающийся на свое описание в словаре. Он должен быть описан как подтип.

Пример представления на языке EXPRESS:

```
*)
ENTITY content_item
ABSTRACT SUPERTYPE;
        dictionary_definition: basic_semantic_unit;
END_ENTITY; -- content_item
(*
```

Определения атрибутов:

dictionary_definition: базовая семантическая единица, используемая для ссылки на определение в словаре.

F.3.4.3 Краткое описание базовых семантических единиц и словарных элементов

Для каждого вида словарных данных должна быть определена пара подтипов **basic_semantic_unit** и **dictionary_element**. На рисунке 5 (как на модели планирования) показаны базовые семантические единицы (БСЕ) и словарные элементы, определенные позже. Отметим, что соотношение между БСЕ и словарным элементом переопределяется для каждого типа данных, так что можно брать только соответствующие пары (на рисунке 5 это не показано).

Каждый вид словарных данных рассматривается в одном из нижеследующих разделов:

- для поставщиков см. F.3.5;
- для классов см. F.3.6;
- для свойств/типов элементов данных см. F.3.7;
- для типов данных см. F.3.8.

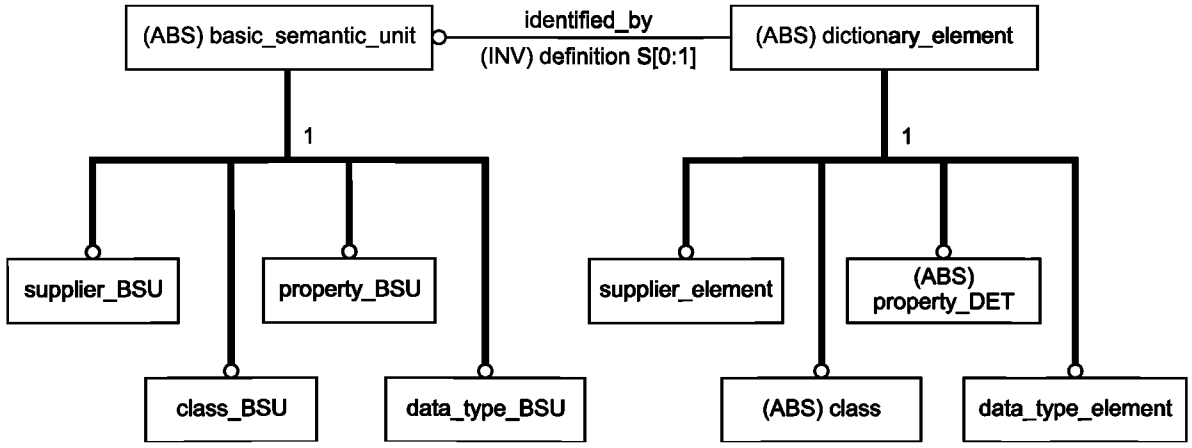


Рисунок F.5 — Текущие БСЕ и словарные элементы

(ABS) basic_semantic_unit	Базовая семантическая единица
identified_by	Идентифицирован …
(INV) definition S[0:1]	Определение …
(ABS) dictionary_element	Словарный элемент
supplier_BSU	БСЕ поставщика
property_BSU	БСЕ свойства
supplier_element	Элемент поставщика
(ABS) property_DET	Тип элемента данных свойства
class_BSU	БСЕ класса

data_type_BSU	БСЕ типа данных
(ABS) class	Класс
data_type_element	Элемент типа данных

F.3.4.4 Идентификация словарных элементов: трехуровневая структура

Абсолютная идентификация базовых семантических единиц основана на использовании нижеследующей трехуровневой структуры:

- поставщик (словарных данных);
- определенный поставщиком словарный элемент (любой определенный поставщиком словарный элемент модели; в данном документе определенный поставщиком словарный элемент — это тип элемента данных свойства **property_DET** и элемент типа данных **data_type_element,** при этом существуют положения, распространяющие данный механизм на другие предметы);
- версия определенного поставщиком словарного элемента.

Абсолютная идентификация может быть получена последовательным включением применимых кодов каждого уровня.

Примечание — Структура указанной абсолютной идентификации отличается от структуры, определенной в МЭК 61360-2 (дублируется для удобства в ИСО 13584-42:1998). В предшествующем издании абсолютная идентификация словарного элемента **dictionary_element,** ассоциированная с областью применения названия **name_scope** (включая **property_DET** и **data_type_element)** составлена так: код поставщика + код класса (соответствующего классу **name_scope**) + код словарного элемента + версия словарного элемента. В данной версии настоящего стандарта код класса удален. Таким образом, код словарного элемента является уникальным для одного типа словарного элемента во всех классах, определенных тем же поставщиком. Для существующих ссылочных словарей официальные органы регистрации, официальные органы технической поддержки и группы стандартизации, отвечающие за стандартные словари, должны гарантировать требуемую единственность (например, путем использования новых кодов в качестве приставки для класса **name_scope)**.

Данная схема идентификации приемлема, если поставщиков несколько. Если в некоторой области приложений приемлемы данные только от одного-единственного поставщика, то некоторые детали идентификации (которые тогда являются константами) могут отсутствовать. Вместе с тем, для выполнения обмена должны быть доступны все уровни во избежание наложения идентификаторов.

Данная схема идентификации формально описана в атрибуте **absolute_id** сущностей **xxx_BSU**, определенных в разделах F.3.5 — F.3.8.

F.3.4.5 Возможности расширения для других типов данных

Механизм словарных элементов БСЕ — очень общий. Он не ограничивается четырьмя видами использованных здесь данных (см. рисунок F.5). Данный раздел указывает некоторые возможности, допускающие расширения других видов данных. В зависимости от того, определена область применения идентификатора класса или поставщик, сущность **xxx_related_BSU** описывается как подтип соответствующим образом. Необходимо также переопределить атрибут идентификации **identified_by** сущности **dictionary_element** (см. разделы 5.7.3—5.10 или текущие виды данных).

F.3.4.5.1 БСЕ, отнесенная к поставщику (Supplier_related_BSU)

Supplier_related_BSU обеспечивает ассоциацию словарного элемента с поставщиком.

Пример — Для серии стандартов ИСО 13584 — это библиотеки программ. Пример представления на языке EXPRESS:

```
*)
ENTITY supplier_related_BSU
ABSTRACT SUPERTYPE
SUBTYPE OF(basic_semantic_unit);
END_ENTITY; -- supplier_related_BSU
(*
```

F.3.4.5.2 БСЕ, отнесенная к классу (Class_related_BSU)

БСЕ, отнесенная к классу (**class_related_BSU),** позволяет ассоциировать словарные элементы с классами.

Пример — Для таблиц и документов ИСО 13584.

Пример представления на языке EXPRESS:

```
*)
ENTITY class_related_BSU
ABSTRACT SUPERTYPE
SUBTYPE OF(basic_semantic_unit);
END_ENTITY; -- class_related_BSU
(*
```

F.3.4.5.3 Соотношение БСЕ поставщика (Supplier_BSU_relationship)

Соотношение **Supplier_BSU_relationship** позволяет ассоциировать БСЕ с поставщиком.

Пример представления на языке EXPRESS:

```
*)
ENTITY supplier_BSU_relationship
ABSTRACT SUPERTYPE;
        relating_supplier: supplier_element;
        related_tokens: SET [1:?] OF supplier_related_BSU;
END_ENTITY; -- supplier_BSU_relationship
(*
```

Определения атрибутов:

relating_supplier: элемент **supplier_element,** идентифицирующий поставщика данных.

related_tokens: набор словарных элементов, ассоциированных с поставщиком, идентифицированным атрибутом **relating_supplier**.

F.3.4.5.4 Соотношение БСЕ класса (Class_BSU_relationship)

Сущность **class_BSU_relationship** позволяет ассоциировать БСЕ с классом.

Пример представления на языке EXPRESS:

```
*)
ENTITY class_BSU_relationship
ABSTRACT SUPERTYPE;
        relating_class: class;
        related_tokens: SET [1:?] OF class_related_BSU;
END_ENTITY; -- class_BSU_relationship
(*
```

Определения атрибутов:

relating_class: класс, идентифицирующий словарный элемент.

related_tokens: набор словарных элементов, ассоциированных с классом, идентифицированным атрибутом **relating_class**.

F.3.5 Поставщик данных

Данный раздел содержит определения для представления данных о самом поставщике. Если поставщиков несколько, то необходимо иметь возможность идентифицировать источник определенного словарного элемента. Рисунок 6 представляет модель планирования данных, ассоциированную с поставщиком с последующим определением EXPRESS.

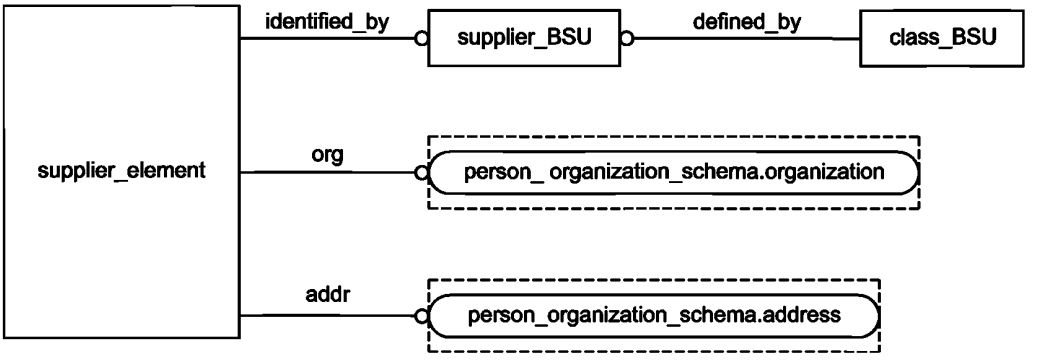


Рисунок F.6 — Краткое описание поставщика данных и соотношений

identified_by	Идентифицирован …
supplier_BSU	БСЕ поставщика
defined_by	Определен …
class_BSU	БСЕ класса
supplier_element	Элемент поставщика
org	Организация
person_ organization_schema.organization	Схема коммерческой организации. Организация
addr	Адрес
person_organization_schema.address	Схема коммерческой организации. Адрес

F.3.5.1 БСЕ поставщика (Supplier_BSU)

Сущность **supplier_BSU** задает уникальную идентификацию поставщиков информации.

Пример представления на языке EXPRESS:

```
*)
ENTITY supplier_BSU
SUBTYPE OF(basic_semantic_unit);
        SELF\basic_semantic_unit.code: supplier_code_type;
DERIVE
        SELF\basic_semantic_unit.version: version_type := '1';
        absolute_id: identifier := SELF\basic_semantic_unit.code;
UNIQUE
        UR1: absolute_id;
END_ENTITY; -- supplier_BSU
(*
```

Определения атрибутов:

code: код поставщика, назначенный в соответствии с ИСО 13584-26.

version: № версии кода поставщика должен быть равен 1.

absolute_id: абсолютная идентификация поставщика.

Пояснения к тексту программы:

UR1: идентификатор поставщика, определенный атрибутом **absolute_id,** является уникальным.

F.3.5.2 Элемент поставщика (Supplier_element)

Сущность **supplier_element** дает словарное описание поставщиков.

Пример представления на языке EXPRESS:

```
*)
ENTITY supplier_element
SUBTYPE OF(dictionary_element);
        SELF\dictionary_element.identified_by: supplier_BSU;
        org: organization;
        addr: address;
INVERSE
        associated_items: SET [0:?] OF supplier_BSU_relationship
                FOR relating_supplier;
END_ENTITY; -- supplier_element
(*
```

Определения атрибутов:

identified_by: БСЕ поставщика **supplier_BSU,** используемая для идентификации рассматриваемого элемента **supplier_element.**

org: организационные данные поставщика.

addr: адрес поставщика.

associated_item: разрешает доступ к другим видам данных с помощью механизма БСЕ.

Пример — Библиотека программ в ИСО 13584-24:2003.

F.3.6 Данные класса

Данный раздел содержит определения для представления словарных данных классов.

F.3.6.1 Общие положения

На рисунке F.7 (в виде модели планирования) представлены данные, ассоциированные с классами, и их соотношения с другими словарными элементами.

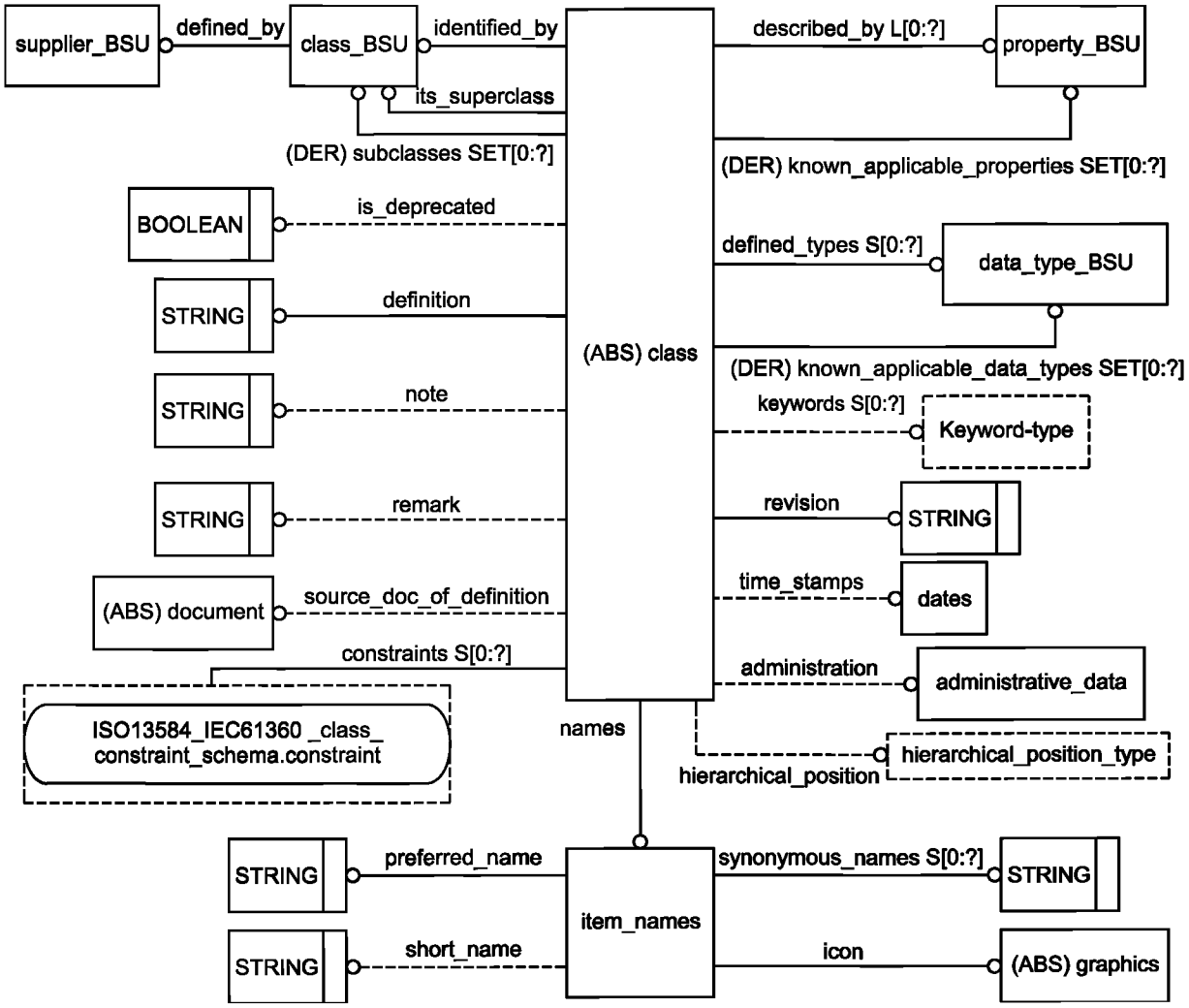


Рисунок F.7 — Краткое описание данных класса и соотношений

supplier_BSU	БСЕ поставщика
defined_by	Определен …
class_BSU	БСЕ класса
identified_by	Идентифицирован …
described_by L[0:?]	Описан …
property_BSU	БСЕ свойства
its_superclass	Его суперкласс
(DER) subclass SET[0:?]	Подклассы
(DER) known_applicable_properties SET[0:?]	Известные применимые свойства
boolean	Булевская переменная
is_deprecated	Больше не используется
defined_types S[0:?]	Определенные типы
data_type_BSU	БСЕ типа данных
string	Строка

definition	Определение
(ABS) class	Класс
(DER) known_applicable_data_types SET[0:?]	Набор известных применимых типов данных
keyword-type	Тип ключевого слова
remark	Заметка
revision	Пересмотр
(ABS) document	Документ
source_doc_of_definition	Исходный документ определения
time_stamps	Временные отметки
dates	Даты
constraints S[0:?]	Ограничения
administration	Администрация
ISO13584_IEC61360 _class _constraint_schema.constraint	Стандартная схема ограничений класса. Ограничение
names	Названия
hierarchical_position_type	Тип положения в иерархии
hierarchical_position	Положение в иерархии
preferred_name	Предпочтительное имя
short_name	Корокое имя
item_names	Названия предметов
synonymous_names S[0:?]	Синонимические имена
(ABS) graphics	Графика
icon	Иконка

Как указано на рисунке F.7, с помощью атрибута **its_superclass** классы формируют дерево наследственности. Важно отметить, что во всем документе термины «наследственность» и «наследовать» используются в соотношении между классами (определенном в словаре). Язык EXPRESS также включает понятие наследственности. Это должно быть четко указано во избежание недоразумений.

Словарные данные классов (см. рисунок F.7) распределены по трем уровням наследственности:

- **class_and_property_element** определяет данные, являющиеся общими и для классов и для типов данных свойств **property_DET**;
- рассматриваемый класс позволяет описывать другие виды классов позже;

Пример — Другие подтипы классов (особенно класс функциональных видов functional_view_class, класс функциональных моделей functional_model_class и класс fm_class_view_of) установлены ИСО 13584-24. Они не характеризуют продукты, но помогают обменивать конкретные представления продуктов (например, геометрические представления).

- класс предметов **item_class** и класс категоризаций **categorization_class** — это сущности, содержащие данные различных классов объектов рассматриваемой области приложения.

Примечание 1 — Два подтипа класса предметов **item_class** (класс компонентов **component_class** и класс материалов **material_class**) определены внутри словарной модели первого издания МЭК 61360. Указанные подтипы больше не используются. Они удалены из настоящего стандарта.

Примечание 2 — Нижеследующие изменения гарантируют, что определения класса словаря, удовлетворяющие требованиям первого издания МЭК 61360-2, удовлетворяют и настоящему изданию: (1) замена классов **component_class** и **material_class** на класс **item_class** с помощью ссылочного словаря; (2) добавление к каждому новому классу предметов **item_class** атрибута совместного использования реализаций **instance_sharable** со значением *true*; (3) добавление к каждому новому классу предметов **item_class** вспомогательного атрибута **hierarchical_position** без задания его значения; (4) добавление к каждому новому классу предметов **item_class** атрибута ключевого слова **keywords**, значением которого является пустое множество.

Примечание 3 — Другой подтип **item_class**, называемый классом особенностей **feature_class**, определен ИСО 13584-24:2003. Данный подтип также больше не используется. Его использование не допускается в новых практических реализациях настоящего стандарта.

Примечание 4 — Нижеследующие изменения гарантируют, что определения класса словаря, удовлетворяющего требованиям ИСО 13584-25, удовлетворяют также и настоящему стандарту: (1) замена класса особенностей **feature_class** на класс предметов **item_class** с помощью ссылочного словаря; (2) добавление к каждому новому классу предметов **item_class** атрибута совместного использования реализаций **instance_sharable** со значением *false*; (3) добавление к каждому новому классу предметов **item_class** вспомогательного атрибута **hierarchical_position** без значения; (4) добавление к каждому новому классу предметов **item_class** атрибута ключевого слова **keywords**, значением которого является пустое множество.

F.3.6.1.1 БСЕ класса (Class_BSU)

Сущность **class_BSU** обеспечивает идентификацию класса.

Пример представления на языке EXPRESS:

```
*)
ENTITY class_BSU
SUBTYPE OF(basic_semantic_unit);
        SELF\basic_semantic_unit.code: class_code_type;
        defined_by: supplier_BSU;
DERIVE
        absolute_id: identifier
                := defined_by.absolute_id + sep_id + dic_identifier;
        known_visible_properties: SET [0:?] OF property_BSU
                := compute_known_visible_properties(SELF);
        known_visible_data_types: SET [0:?] OF data_type_BSU
                := compute_known_visible_data_types (SELF);
INVERSE
        subclasses: SET [0:?] OF class FOR its_superclass;
        added_visible_properties: SET [0:?] OF property_BSU
                FOR name_scope;
        added_visible_data_types: SET [0:?] OF data_type_BSU
                FOR name_scope;
UNIQUE
        UR1: absolute_id;
END_ENTITY; -- class_BSU
(*
```

Определения атрибутов:

code: код, назначенный данному классу его поставщиком.

defined_by: поставщик, определяющий данный класс и его словарный элемент.

absolute_id: уникальная идентификация данного класса.

known_visible_properties: набор БСЕ свойств **property_BSU,** ссылающихся на класс как атрибут имени **name_scope,** или на любой известный суперкласс данного класса, и, следовательно, являющихся видимыми в данном классе (любом его подклассе).

Примечание 1 — Если словарное определение **dictionary_definition** некоторого класса не присутствует в рассматриваемом контексте обмена (а контекст обмена PLIB никогда не предполагается полным), то суперкласс для рассматриваемого класса может быть не известен. Следовательно, свойства, определенные как видимые в данном суперклассе, не принадлежат атрибуту **known_visible_properties**. Только для получающей системы все словарные определения **dictionary_definition** БСЕ должны быть доступными. Следовательно, для получающей системы атрибут **known_visible_properties** содержит все свойства, видимые в данном классе.

known_visible_data_types: набор БСЕ типов данных **data_type_BSU,** ссылающихся на класс как атрибут **name_scope** или на любой известный суперкласс данного класса, и, следовательно, являющихся видимыми в данном классе (любом его подклассе).

Примечание 2 — Если некоторые словарные определения **dictionary_definition** класса не присутствуют в рассматриваемом контексте обмена (а контекст обмена библиотеки PLIB никогда не предполагается полным), то суперкласс для данного класса может быть не известен. Следовательно, типы данных **data_types**, определенные как видимые данным суперклассом, не принадлежат атрибуту известного видимого типа данных **known_visible_data_type**. Для получающей системы все словарные определения **dictionary_definition** БСЕ должны быть доступными. Следовательно, для получающей системы, атрибут **known_visible_data_type** содержит все типы данных **data_types**, видимые в данном классе.

subclasses: набор классов, определяющих данный класс как их суперкласс.

added_visible_properties: набор БСЕ свойств **property_BSU,** ссылающихся на класс как **name_scope**, и, следовательно, являющихся видимыми в данном классе (любом его подклассе).

Примечание 3 — Данный атрибут ссылается только на БСЕ свойств **property_BSU**, принадлежащих рассматриваемому контексту обмена. Для получающей системы они могут уже завершить выполнение других БСЕ свойств **property_BSU**, ссылающихся на данный класс (контекст обмена библиотеки PLIB никогда не предполагается полным).

Примечание 4 — Атрибут **added_visible_properties** используется для кодирования атрибута «Видимые свойства» класса.

added_visible_data_type: набор БСЕ типов данных **data_type_BSU,** ссылающихся на класс как **name_scope** и, следовательно, являющихся видимыми в данном классе (любом его подклассе).

Примечание 5 — Данный обратный атрибут ссылается только на БСЕ типов данных **data_type_BSU**, принадлежащих рассматриваемому контексту обмена. Для получающей системы они могут уже завершить выполнение других **data_type_BSU**, ссылающихся на данный класс (контекст обмена библиотеки PLIB никогда не предполагается полным).

Примечание 6 — Атрибут **added_visible_data_type** используется для кодирования атрибута «Видимые типы».

Пояснения к тексту программы:
UR1: последовательность кода поставщика и кода класса является уникальной.
F.3.6.1.2 Класс и элемент свойства (Class_and_property_element)
Сущность **class_and_property_element** содержит атрибуты, являющиеся общими и для классов и для БСЕ свойств **property_DET.**
Пример представления на языке EXPRESS:

```
        *)
        ENTITY class_and_property_elements
        ABSTRACT SUPERTYPE OF(ONEOF(
                property_DET,
                class))
        SUBTYPE OF(dictionary_element);
                names: item_names;
                definition: definition_type;
                source_doc_of_definition: OPTIONAL document;
                note: OPTIONAL note_type;
                remark: OPTIONAL remark_type;
        END_ENTITY; -- class_and_property_elements
        (*
```

Определения атрибутов:
names: названия, описывающие данный словарный элемент.
definition: текст, описывающий данный словарный элемент.
source_doc_of_definition: исходный документ для данного текстового описания.
note: дополнительная информация о любой части словарного элемента, существенной для понимания.
remark: дополнительный текст, поясняющий смысл данного словарного элемента.

Примечание 1 — Атрибут **names** используется как отправная точка при кодировании (в сущности **item_names**) свойства и атрибутов «Предпочтительное имя», «Короткое имя», «Синонимическое имя».

Примечание 2 — Атрибут определения **definition** используется для кодирования атрибута свойства «Определение» и атрибута класса «Определение».

Примечание 3 — Атрибут исходного документа для определения **source_of_doc_definition** используется для кодирования атрибута свойства «Исходный документ определения» и атрибута класса «Исходный документ определения».

Примечание 4 — Атрибут примечания **note** используется для кодирования атрибутов свойств и классов «Примечание».

Примечание 5 — Атрибут заметки **remark** используется для кодирования атрибутов свойств и классов «Заметка».

F.3.6.1.3 Класс (class)
Сущность **class** — это абстрактный ресурс для всех видов классов.
Пример представления на языке EXPRESS:

```
    *)
    ENTITY class
```

```
ABSTRACT SUPERTYPE OF( ONEOF (item_class, categorization_class))
SUBTYPE OF(class_and_property_elements);
        SELF\dictionary_element.identified_by: class_BSU;
        its_superclass: OPTIONAL class_BSU;
        described_by: LIST [0:?] OF UNIQUE property_BSU;
        defined_types: SET [0:?] OF data_type_BSU;
        constraints: SET [0:?] OF constraint_or_constraint_id;
        hierarchical_position: OPTIONAL hierarchical_position_type;
        keywords: SET [0:?] OF keyword_type;
        sub_class_properties: SET [0:?] OF property_BSU;
        class_constant_values: SET [0:?] OF class_value_assignment;
DERIVE
        subclasses: SET [0:?] OF class := identified_by.subclasses;
        known_applicable_properties: SET [0:?] OF property_BSU
                := compute_known_applicable_properties(
                        SELF\dictionary_element.identified_by);
        known_applicable_data_types: SET [0:?] OF data_type_BSU
                := compute_known_applicable_data_types(
                        SELF\dictionary_element.identified_by);
        known_property_constraints: SET [0:?] OF property_constraint
                := compute_known_property_constraints(
                [SELF\dictionary_element.identified_by]);
INVERSE
        associated_items: SET [0:?] of class_BSU_relationship
                FOR relating_class;
WHERE
        WR1: acyclic_superclass_relationship (SELF.identified_by, [ ]);
        WR2: NOT all_class_descriptions_reachable (
                SELF\dictionary_element.identified_by)
                OR (list_to_set(SELF.described_by) <=
                SELF\dictionary_element.identified_by
                \class_BSU.known_visible_properties);
        WR3: NOT all_class_descriptions_reachable (
                SELF\dictionary_element.identified_by)
                OR (SELF.defined_types <=
                SELF\dictionary_element.identified_by
                \class_BSU.known_visible_data_types);
        WR5: NOT all_class_descriptions_reachable (
                SELF\dictionary_element.identified_by)
                OR (QUERY (cdp <* described_by
                | (SIZEOF (cdp\basic_semantic_unit.definition)=1)
                AND (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
                +'.DEPENDENT_P_DET') IN TYPEOF
                (cdp\basic_semantic_unit.definition [1]))
                AND NOT
                (cdp\basic_semantic_unit.definition [1].depends_on
                <= known_applicable_properties))=[ ]);
        WR6: check_datatypes_applicability(SELF);
        WR7: QUERY (cons <* constraints
                | ('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
                +'.INTEGRITY_CONSTRAINT' IN TYPEOF (cons))
                AND (SIZEOF (cons\property_constraint.constrained_property
                .definition) =1)
                AND NOT correct_constraint_type (
                cons\integrity_constraint.redefined_domain,
                cons\property_constraint.constrained_property
                .definition [1].domain)) = [ ];
        WR8: QUERY (cons <* constraints
                | (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
                +'.CONFIGURATION_CONTROL_CONSTRAINT') IN TYPEOF (cons))
                AND NOT correct_precondition (cons, SELF)) = [ ];
        WR9: NOT all_class_descriptions_reachable (
                SELF\dictionary_element.identified_by)
                OR (QUERY (cons <* constraints
```

```
                I (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
                +'.PROPERTY_CONSTRAINT') IN TYPEOF (cons))
                AND NOT
                ((cons\property_constraint.constrained_property
                IN SELF\dictionary_element.identified_by
                \class_BSU.known_visible_properties)
                OR (cons\property_constraint.constrained_property
                IN known_applicable_properties)))=[ ]);
        WR10: (SIZEOF( QUERY (lab <* keywords
                I ('ISO13584_IEC61360_DICTIONARY_SCHEMA'
                +'.LABEL_WITH_LANGUAGE') IN TYPEOF (lab)))
                = SIZEOF( keywords))
                OR (SIZEOF (QUERY (lab <* keywords
                I ('ISO13584_IEC61360_DICTIONARY_SCHEMA'
                +'.LABEL_WITH_LANGUAGE') IN TYPEOF (lab)))
                = SIZEOF( keywords));
        WR11: (('ISO13584_IEC61360_ITEM_CLASS_CASE_OF_SCHEMA'
                + '.A_PRIORI_SEMANTIC_RELATIONSHIP')
                IN TYPEOF (SELF)) OR
                (QUERY(p <* sub_class_properties
                I NOT(p IN SELF.described_by)) = [ ]);
        WR12: NOT all_class_descriptions_reachable (SELF.identified_by) OR
                (QUERY (va <* class_constant_values |
                NOT is_class_valued_property (
                va.super_class_defined_property, SELF.identified_by)) = [ ]);
        WR13: QUERY(val <* SELF.class_constant_values
                I QUERY (v <* class_value_assigned (
                val.super_class_defined_property, SELF.identified_by)
                I val.assigned_value <> v) <>[ ]) = [ ];
  END_ENTITY; -- class
  (*
```

Определения атрибутов:

identified_by: БСЕ класса **class_BSU,** задающая данный класс.

its_superclass: ссылка на класс, для которого рассматриваемый класс является подклассом.

described_by: список ссылок на дополнительные свойства, доступные для использования в описании продуктов внутри класса и любого его подкласса.

Примечание 1 — Свойство может также быть применимым в классе, если оно импортировано из другого класса с помощью априонного семантического соотношения **a_priori_semantic_relationship** (см. раздел F.6). Следовательно, свойства, на которые производится ссылка атрибутом **described_by**, не определяют все применимые свойства класса.

Примечание 2 — Списочный порядок — это порядок представления свойств, предложенный поставщиком.

Примечание 3 — Свойство, являющееся контекстно-зависимым (**context_dependent_P_DET**), может стать применимым в классе, только если все контекстные параметры (**condition_DET**), от которых зависит его значение, также являются применимыми в данном классе. См. ниже Правило 5 (WR5).

defined_types: набор ссылок на типы, которые могут быть использованы для различных типов элементов данных свойств **property_DET** с помощью дерева наследственности, нисходящего из данного класса.

Примечание 4 — Тип данных **data_type** также может быть применимым в классе, если этот тип **data_type** импортирован из другого класса с помощью априорного семантического соотношения **a_priori_semantic_relationship** (см. раздел F.6). Следовательно, типы данных, на которые производится ссылка атрибутом **defined_type**, не определяют все применимые типы данных в классе.

constraints: набор ограничений, ограничивающих целевые области значений некоторых свойств класса до некоторых подмножеств унаследованных областей значений.

Примечание 5 — Каждое ограничение атрибута **constraints** должно выполняться для реализаций класса. Таким образом, атрибут **constraints** — это сопряжение ограничений.

hierarchical_position: кодированное представление положения класса в иерархии включения классов, которой он принадлежит; иерархическая позиция **hierarchical_position** класса изменяется, если изменяется структура класса онтологии. Таким образом, данный атрибут не может использоваться как устойчивый идентификатор класса.

Примечание 6 — Данный вид кодированного имени используется, в особенности, в иерархии категоризаций продукта для представления структуры включения классов с помощью некоторых конвенций кодирования.

Пример 1 — В спецификации UNSPSC ООН Производственные Компоненты и Поставки имеют иерархическое положение 31000000. Металлические крепежные изделия занимают иерархическое положение 31160000. Болты занимают иерархическое положение 31161600. По конвенции данное представление иерархического положения допускает, чтобы Производственные Компоненты и Поставки располагались на первом уровне иерархии, Металлические крепежные изделия располагались на втором уровне иерархии (и включались в Производственные Компоненты и Поставки), а Болты были на третьем уровне иерархии (и включались в Металлические крепежные изделия).

keywords: набор ключевых слов (возможно, на нескольких языках), позволяющих отыскивать нужный класс.

sub_class_properties: объявляет свойство, как имеющее значение класса, т. е. в нескольких подклассах его одно-единственное значение назначено для всего класса (см. F 3.7.4).

class_constant_value: назначения в текущем классе для свойств, имеющих значение класса и объявленных в суперклассах (см. F.3.7.4).

subclasses: набор классов, указывающих данный класс как их суперкласс.

known_applicable_properties: БСЕ свойства **property_BSU,** на которые производится ссылка некоторым классом или любым его известным суперклассом с помощью атрибута **described_by,** и которые, следовательно, являются применимыми в данном классе (любом его подклассе).

Примечание 7 — Если некоторое словарное определение **dictionary_definition** класса отсутствует в рассматриваемом контексте обмена (а контекст обмена библиотеки PLIB никогда не предполагается полным), то суперкласс некоторого класса может быть не известен. Следовательно, свойства, определенные как применимые в данном суперклассе, не принадлежат атрибуту известных применимых свойств **known_applicable_properties**. Все словарные определения **dictionary_definition** БСЕ должны быть доступны только для получающей системы. Следовательно, для получающей системы атрибут **known_applicable_properties** содержит все свойства, являющиеся применимыми в классе, так как на них производится ссылка атрибутом описания **described_by**.

known_applicable_data_type: БСЕ типа данных **data_type_BSU,** на которые производится ссылка из класса или любого его известного суперкласса с помощью атрибута **defined_types,** и которые, следовательно, являются применимыми в данном классе (любом его подклассе).

Примечание 8 — Если некоторое словарное определение **dictionary_definition** класса отсутствует в рассматриваемом контексте обмена (а контекст обмена библиотеки PLIB никогда не предполагается полным), то суперкласс некоторого класса может быть не известен. Следовательно, типы данных **data_types**, определенные как применимые в данном суперклассе, не принадлежат атрибуту **known_applicable_data_type**. Только для получающей системы все словарные определения **dictionary_definition** БСЕ должны быть доступными. Следовательно, для получающей системы атрибут **known_applicable_data_types** содержит все типы данных **data_types**, являющиеся применимыми в классе, так как на них производится ссылка атрибутом определения типа **defined_types**.

known_property_constraint: ограничения **constraints** на свойство, на которые производится ссылка некоторым классом или любой его известным представительным суперклассом с помощью атрибута **constraints,** или, в случае класса, являющегося подтипом априорного семантического соотношения **a_priori_semantic_relationship,** с помощью атрибута ссылочного ограничения **referenced_constraints**.

associated_item: дает доступ к другим видам данных с помощью БСЕ механизма.

Пояснения к тексту программы:

WR1: наследственная структура, определенная иерархией классов, не содержит циклы.

WR2: только свойства, являющиеся видимыми в данном классе, могут стать применимыми в данном классе, так как на них производится ссылка атрибутом описания **described_by**.

WR3: только типы данных, являющиеся видимыми в данном классе, могут стать применимыми в данном классе, так как на них производится ссылка атрибутом **defined_types**.

WR4: только свойства, не являющиеся применимыми в классе по наследству, могут стать применимыми в данном классе, так как на них производится ссылка атрибутом описания **described_by**.

WR5: только контекстно-зависимые свойства **(dependent_P_DET),** все контекстные параметры **(condition_DET)** которых являются применимыми в классе, могут стать применимыми в данном классе, так как на них производится ссылка атрибутом описания **described_by**.

WR6: только типы данных, не являющиеся применимыми в классе по наследству, могут стать применимыми в данном классе, так как на них производится ссылка атрибутом определения типа **defined_types**.

Примечание 9 — Атрибут **its_superclass** используется для кодирования атрибута класса «Суперкласс».

Примечание 10 — Атрибут **described_by** обеспечивает кодирование «Применимых свойств» класса.

Примечание 11 — Атрибут **defined_type** используется для кодирования атрибута «Применимых типов» класса.

WR7: набор ограничений, являющихся ограничениями свойств, должен определять ограничения, совместимые с областью значений свойств, где последние применимы.

WR8: все свойства, на которые производится ссылка в предварительном условии ограничения управления конфигурацией **configuration_control_constraint,** должны быть применимыми в классе.

WR9: все свойства, на которые производится ссылка в атрибуте ограничения **constraint**, должны быть либо видимыми, либо применимыми в классе.

WR10: либо все ключевые слова **keywords** представлены как метки с языком **label_with_languages**, либо все они представлены как просто метки **labels.**

WR11: если класс не является априорным семантическим соотношением **a_priori_semantic_relationship,** то свойства подкласса **sub_class_properties** должны принадлежать списку **described_by**.

Примечание 12 — Свойства подкласса **sub_class_properties** могут быть также импортированы с помощью априорного семантического соотношения **a_priori_semantic_relationship**.

WR12: Свойства, на которые производится ссылка в атрибуте постоянных значений класса **class_constant_value,** объявляются как имеющие значение класса в некотором суперклассе текущего класса или в самом текущем классе.

Примечание 13 — Атрибут свойств подкласса **sub_class_properties** сущности **class** используется для кодирования атрибута «Свойств со значением класса» для классов.

Примечание 14 — Атрибут постоянных значений класса **class_constant_value** сущности **class** используется для кодирования «Постоянных значений класса» для классов.

WR13: если свойству, на которое производится ссылка из области **class_constant_value**, уже назначено значение в суперклассе, то его значение, назначенное в текущем классе, должно быть тем же самым.

Дополнительные пояснения:

IP1: если все представительные суперклассы рассматриваемого класса доступны, то известные ограничения свойства **known_property_constraints** — это ограничения свойств, присоединенных к данному классу либо как видимые, либо как применимые свойства.

F.3.6.2 Класс предметов (item_class)

Сущность **item_class** позволяет моделировать любой тип сущности области приложения, который может быть отнесен к характеристическому классу, определенному структурой класса и набором свойств. Как реализации продуктов, так и реализации конкретных аспектов продуктов, представленные как особенности, отображаются на **item_class**.

Сущность **item_class** включает атрибут совместного использования реализации **instance_sharable,** указывающий концептуальный статус предмета. Если значение данного атрибута *true,* то каждая реализация представляет собой независимый предмет. В противном случае — это особенность, т. е. зависимый предмет, являющийся компонентом другого предмета. Вышесказанное не предписывает какой-либо особой практической реализации на уровне представления данных.

Пример — Головка винта — это особенность, описанная рядом свойств. Она может существовать, только если на нее производится ссылка сущностью винт. Она представляется как класс предметов item_class с атрибутом instance_sharable, равным false.

Примечание 1 — Внутри словарной модели первого издания ИСО 13584-42 и МЭК 61360-2 определено два подтипа класса предметов **item_class**: класс компонентов **component_class** и класс материалов **material_class**. Указанные подтипы больше не используются, они удалены из данного издания МЭК 61360.

Примечание 2 — Другой подтип **item_class**, названный классом особенностей **feature_class**, определен в ИСО 13584-24:2003. Данный подтип также больше не используется. Его использование не рекомендуется в новых практических реализациях настоящего стандарта.

Пример представления на языке EXPRESS:

```
*)
ENTITY item_class
SUBTYPE OF(class);
        simplified_drawing: OPTIONAL graphics;
        coded_name: OPTIONAL value_code_type;
        instance_sharable: OPTIONAL BOOLEAN;
END_ENTITY; -- item_class
(*
```

Определения атрибутов:

simplified_drawing: вспомогательные рисунки (графика), ассоциированные с описанным классом.

Примечание 3 — Атрибут **simplified_drawing** сущности **item_class** используется для кодирования атрибута «Упрощенный рисунок» для классов.

coded_name: может использоваться как постоянное значение класса для характеристики данного класса в области значений свойств **sub_class_properies** соответствующего суперкласса.

Примечание 4 — В серии стандартов ИСО 13584 данный атрибут не используется. Он используется только в серии стандартов МЭК 61360.

instance_sharable: если значение равно *false*, то реализации класса предметов **item_class** являются особенностями; если значение не определено или равно *true,* то реализации **item_class** являются независимыми предметами.

Примечание 5 — В общей словарной модели ИСО 13584/МЭК 61360 от практической реализации зависит принятие решения: действительно ли несколько реализаций особенностей, моделируемых рассматриваемым набором пар «свойство-значение», представлены несколькими блоками данных EXPRESS, или они представлены этим блоком данных в файле обмена данных. Таким образом, реализация класса предметов **item_class** со значением атрибута **instance_sharable** равным *false*, на которую производится ссылка несколькими реализациями **item_class** на уровне модели данных, интерпретируется как несколько реальных реализаций рассматриваемой особенности.

F.3.6.3 Класс категоризаций (Categorization_class)

Сущность **categorization_class** позволяет моделировать образование групп из набора объектов, составляющих элемент категоризации.

Пример 1 — Производственные компоненты и поставки и промышленная оптика — это примеры класса категоризации продуктов, определенного UNSPSC.

Ни свойства, ни типы данных, ни ограничения не ассоциируются (как видимые или применимые) с таким классом. Более того, классы категоризаций **categorization_classes** не могут быть отнесены друг к другу с помощью представительного наследственного соотношения. Они могут быть отнесены друг к другу только с помощью условного соотношения между классами. Особый атрибут, называемый **categorization_class_superclass,** позволяет занести классы категоризаций **categorization_classes,** являющиеся суперклассами **categorization_class,** в условную иерархию.

Примечание — С помощью условных соотношений конструктивы ресурса, классы предметов **item_class** могут также быть соединены с классами категоризаций **categorization_class.**

Пример 2 — Нижеследующий пример показывает, как характеристические классы и классы категоризаций могут быть соединены для достижения некоторых конкретных целей. Поставщик шариковых подшипников разрабатывает свою собственную онтологию, что облегчает получение и использование продукта. Для достижения указанной цели, он использует стандартные свойства и пытается использовать стандартную классификацию. Он поставляет только шариковые подшипники: некоторые подшипники герметичны, а некоторые — нет. Определенные свойства могут быть ассоциированы только с герметичными подшипниками, а с негерметичными — нет. Однако указанные категории не существуют как классы в стандартной онтологии подшипников. Поэтому поставщик подшипников поступает следующим образом. (1) Он разрабатывает собственную онтологию, включающую три характеристических класса: my_bearing (мои подшипники), my_sealed_bearings (мои герметичные подшипники), my_non_sealed_bearing (мои негерметичные подшипники). Два последних класса соединены с первым представительным наследственным соотношением, и все свойства, назначенные первому, унаследуются последними. (2) Чтобы использовать некоторые свойства, определенные ИСО/ТС 23768-1 (находится в разработке), поставщик подшипников указывает, что класс my_bearings — это условный стандартный класс подшипников ball bearing, определенный в ИСО/ТС 23768-1. С помощью данного условного соотношения поставщик может импортировать (в свой класс my_bearings) стандартные свойства: внутренний диаметр, наружный диаметр, класс допуска ИСО. Более того, он создает необходимые свойства, не определенные в стандарте. (3) Для получения сервера, визуализирующего каталог поставщика, представляется небольшой фрагмент классификации UNSPSC, а также условное соотношение между стандартным классом ball_bearings UNSPSC и своим собственным классом my_bearing. Пример дан ниже на рисунке F.8:

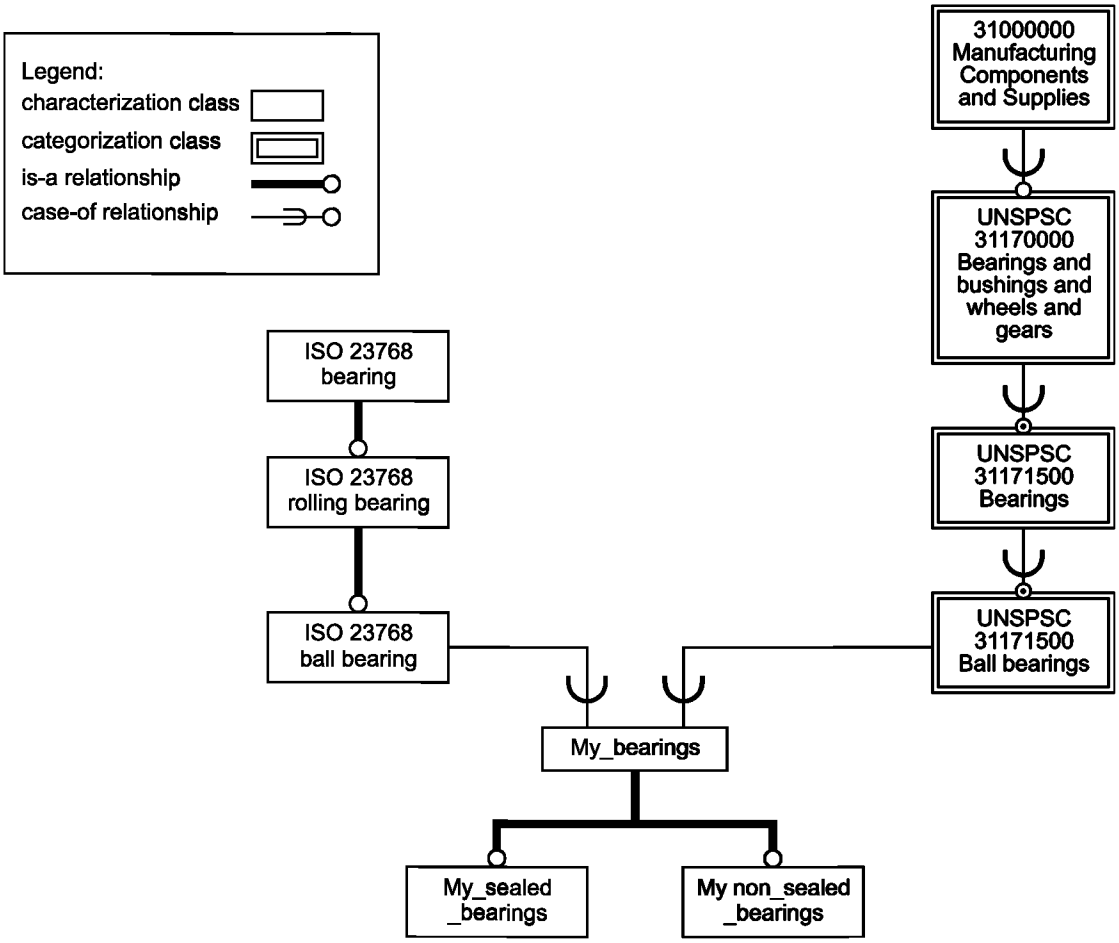


Рисунок F.8 — Пример онтологии поставщика

legend	Легенда
characterization class	Характеристический класс
categorization class	Класс категоризаций
Is-a relationship	Представительное соотношение
Case-of relationship	Условное соотношение
31000000 Manufacturing Components and Supplies	Код по классификации ООН 31000000 Производственные компоненты и поставки
UNSPSC 31170000 Bearings and bushings and wheels and gears	Подшипники, втулки, колеса и зубчатые передачи
ISO 23768 bearing	Подшипник
ISO 23768 rolling bearing	Роликовый подшипник
UNSPSC 31171500 Bearings	Подшипники
ISO 23768 ball bearings	Шариковый подшипник
My_bearings	Класс «мои подшипники»
UNSPSC 31171504 Ball bearings	Шариковые подшипники по классификации ООН
My_sealed_bearings	Класс «мои герметичные подшипники»
My non_sealed_bearings	Класс «мои негерметичные подшипники»

Пример представления на языке EXPRESS

```
*)
ENTITY categorization_class
SUBTYPE OF(class);
        categorization_class_superclasses: SET [0:?] of class_BSU;
WHERE
        WR1: QUERY (cl <* SELF. categorization_class_superclasses
                | NOT (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
                +'.CATEGORIZATION_CLASS') IN TYPEOF(cl.definition[1])))
                = [ ];
        WR2: NOT EXISTS(SELF\class.its_superclass);
        WR3: SIZEOF(SELF\class.described_by) = 0;
        WR4: SIZEOF(SELF\class.defined_types) = 0;
        WR5: SIZEOF(SELF\class.constraints) = 0;
        WR6: SIZEOF(compute_known_visible_properties
                (SELF\dictionary_element.identified_by)) = 0;
        WR7: SIZEOF(SELF\class.sub_class_properties) = 0;
        WR8: SIZEOF(SELF\class.class_constant_values) = 0;
        WR9: SIZEOF(SELF\class.identified_by.known_visible_properties)
                = 0;
        WR10: SIZEOF(SELF\class.identified_by.known_visible_data_types)
                = 0;
END_ENTITY; -- categorization_class
(*
```

Определения атрибутов:

categorization_class_superclasses: классы категоризаций **categorization_class,** расположенные на одну ступень выше данного класса категоризаций в условной иерархии классов.

Пояснения к тексту программы:

WR1: только классы категоризаций **categorization_class** могут быть суперклассами для **categorization_class.**

WR2: categorization_class не должен иметь представительного суперкласса.

WR3: свойства не могут быть ассоциированы с **categorization_class.**

WR4: типы данных не могут быть ассоциированы с **categorization_class.**

WR5: ограничения не могут быть ассоциированы с **categorization_class.**

WR6: categorization_class не может быть классом определения свойства для какого-либо свойства.

WR7: свойство подкласса не может быть ассоциировано с **categorization_class.**

WR8: постоянное значение класса не может быть ассоциировано с **categorization_class.**

WR9: видимое свойство не может быть ассоциировано с **categorization_class.**

WR10: видимый тип данных не может быть ассоциирован с **categorization_class.**

F.3.7 Тип элемента данных/данные для свойств

Данный раздел содержит определения словарных данных для свойств.

F.3.7.1 БСЕ свойства (Property_BSU)

Сущность **property_BSU** дает идентификацию свойства.

Пример представления на языке EXPRESS:

```
*)
ENTITY property_BSU
SUBTYPE OF(basic_semantic_unit);
        SELF\basic_semantic_unit.code: property_code_type;
        name_scope: class_BSU;
DERIVE
        absolute_id: identifier :=
                name_scope.defined_by.absolute_id
                + sep_id + dic_identifier;
INVERSE
        describes_classes: SET OF class FOR described_by;
UNIQUE
        UR1: absolute_id;
WHERE
        WR1: QUERY(c <* describes_classes |
                NOT (is_subclass (c, name_scope.definition [1])))= [ ];
END_ENTITY; -- property_BSU
(*
```

Определения атрибутов:

code: данный код необходим для уникальной идентификации свойства во всех онтологиях, определенных рассматриваемым поставщиком с атрибутом **name_scope.defined_by**.

name_scope: ссылка на класс, на котором или ниже которого рассматриваемый элемент свойства доступен для ссылки атрибутом описания **described_by.**

absolute_id: уникальная идентификация данного свойства.

describes_classes: классы, объявляющие данное свойство доступным для использования в описании продукта.

Пояснения к тексту программы:

WR1: любой класс, на который производится ссылка атрибутом **describes_class** БСЕ свойства **property_ BSU,** либо это класс, на который производится ссылка его атрибутом **name_scope,** либо это подкласс данного класса.

UR1: идентификатор свойства **absolute_id** является уникальным.

Примечание — Атрибут **name_scope** сущности **property_BSU** используется для кодирования атрибута «Класс определений» для свойств (см. 7.2.2).

F.3.7.2 Типы элементов данных свойств (Property_DET)

Сущность **property_DET** дает словарные описания свойств.

Пример представления на языке EXPRESS:

```
*)
ENTITY property_DET
ABSTRACT SUPERTYPE OF(ONEOF(
    condition_DET, dependent_P_DET, non_dependent_P_DET))
SUBTYPE OF(class_and_property_elements);
    SELF\dictionary_element.identified_by: property_BSU;
    preferred_symbol: OPTIONAL mathematical_string;
    synonymous_symbols: SET [0:?] OF mathematical_string;
    figure: OPTIONAL graphics;
    det_classification: OPTIONAL DET_classification_type;
    domain: data_type;
    formula: OPTIONAL mathematical_string;
DERIVE
    describes_classes: SET [0:?] OF class
    := identified_by.describes_classes;
END_ENTITY; -- property_DET
(*
```

Определения атрибутов:

identified_by: БСЕ свойства **property_BSU, идентифицирующая данное свойство.**

preferred_symbol: более короткое описание данного свойства.

synonymous_symbols: синоним для более короткого описания свойств.

figure: вспомогательная графика для описания свойства.

det_classification: класс ИСО 80000/МЭК 80000 (ранее ИСО 31) для данного свойства.

domain: ссылка на тип данных **data_type,** ассоциированный со свойством.

formula: математическое выражение, объясняющее свойство.

describes_classes классы, объявляющие данное свойство доступным для использования в описании продукта.

Примечание 1 — Атрибут **preferred_symbol** используется для кодирования атрибута «Предпочтительный буквенный символ» для свойств.

Примечание 2 — Атрибут **synonymous_symbols** используется для кодирования атрибутов «Синонимических буквенных символов» для свойств.

Примечание 3 — Атрибут **det_classification** используется для кодирования атрибута «Классификация типов свойств».

Примечание 4 — Атрибут области **domain** используется как отправная точка для кодирования атрибута свойства «Тип данных». Сущность **data_type** может быть описана как подтип для различных возможных типов данных.

Примечание 5 — Атрибут **formula** используется для кодирования атрибута «Формула» для свойств.

На рисунке F.9 дана модель планирования данных, ассоциированных с сущностями **property_DETs**

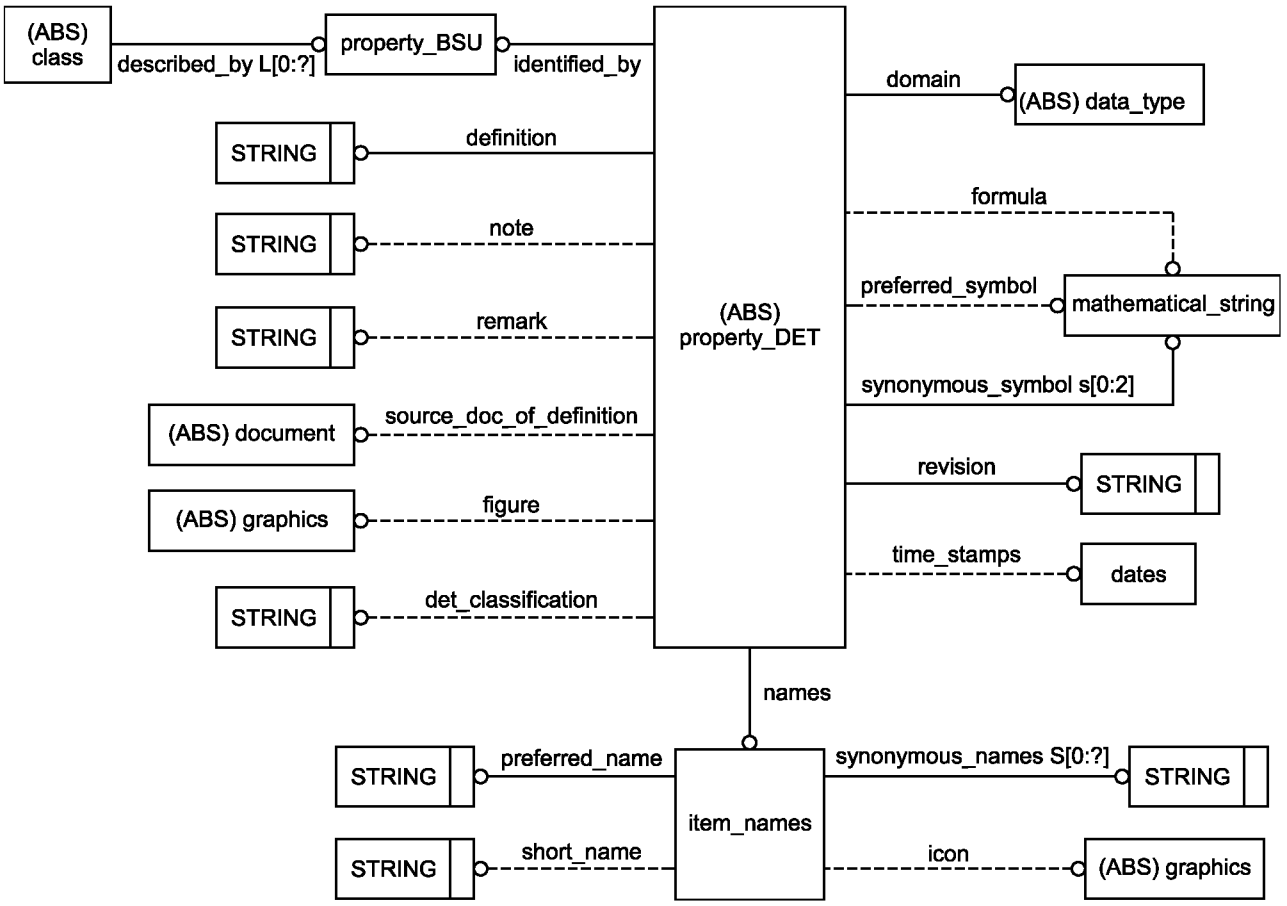


Рисунок F.9 — Краткое описание свойств типов элементов данных и соотношений

(ABS) class	Класс
described_by L[0:?]	Описан …
property_BSU	БСЕ свойства
identified_by	Идентифицирован …
domain	Область
(ABS) data_type	Тип данных
string	Строка
definition	Определение
formula	Формула
note	Примечание
remark	Заметка
(ABS) property_DET	Тип элемента данных свойства
preferred_symbol	Предпочтительный символ
mathematical_string	Математическая строка
(ABS) document	Документ
source_doc_of_definition	Исходный документ определения

synonymous_symbol s[0:2]	Синонимический символ
revision	Пересмотр
(ABS) graphics	Графика
figure	Рисунок
time_stamps	Отметки времени
dates	Даты
det_classification	Классификация типов элементов данных
preferred_name	Предпочтительное имя
names	Имена
synonymous_names S[0:?]	Синонимическое имя
short_name	Короткое имя
item_names	Имена предметов
icon	Иконка

F.3.7.3 Условные, зависимые и независимые типы элементов данных

На рисунке F.10 даны различные виды типов элементов данных в формате модели планирования.

Отметим, что рисунок F.10 упрощен. Вообще говоря, соотношение **«depends_on»** должно применяться со ссылкой на БСЕ. Но установлено ограничение, по которому ссылочный тип элемента данных свойства **property_DET** должен быть типом элемента данных условия **condition_DET** (см. сущность **dependent_P_DET**).

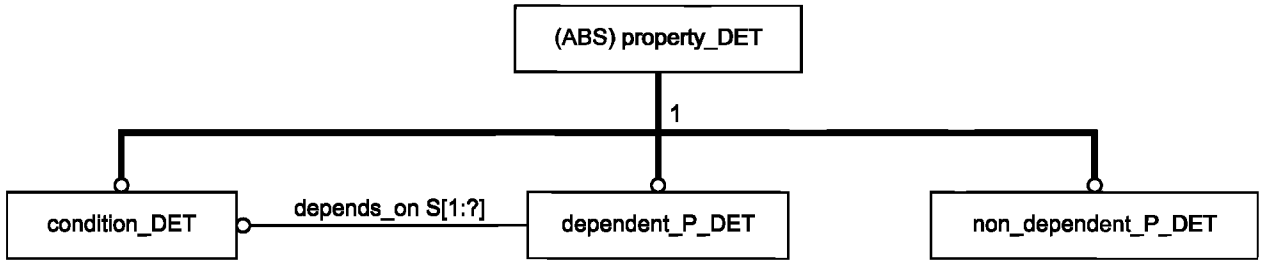


Рисунок F.10 — Виды типов элементов данных

(ABS) property_DET	Тип элемента данных свойства
condition_DET	Тип элемента данных условия
depends_on S[1:?]	Зависит от ...
dependent_P_DET	Зависимый тип элемента данных свойства
non_dependent_P_DET	Независимый тип элемента данных свойства

F.3.7.3.1 Тип элементов данных условия (Condition_DET)

Сущность **condition_DET** — это свойство, от которого могут зависеть другие свойства.

Пример представления на языке EXPRESS:

```
*)
ENTITY condition_DET
SUBTYPE OF(property_DET);
END_ENTITY; -- condition_DET
(*
```

F.3.7.3.2 Зависимый тип элемента данных свойства (Dependent_P_DET)

Сущность **dependent_P_DET** — это свойство, значение которого явно зависит от значений некоторых условий.

Пример — Сопротивление термистора зависит от наружной температуры. Сопротивление термистора должно быть представлено как dependent_P_DET, а наружная температура термистора — это condition_DET.

Пример представления на языке EXPRESS:

```
*)
ENTITY dependent_P_DET
SUBTYPE OF (property_DET);
    depends_on: SET [1:?] OF property_BSU;
WHERE
    WR1: QUERY (p <* depends_on | NOT (definition_available_implies (
            p, ('ISO13584_IEC61360_DICTIONARY_SCHEMA.CONDITION_DET'
            IN TYPEOF(p.definition [1])))) = [ ];
END_ENTITY; -- dependent_P_DET
(*
```

Определения атрибутов:

depends_on: набор базовых семантических единиц, идентифицирующих свойства, от которых зависит данное свойство.

Пояснения к тексту программы:

WR1: только условные типы элементов данных **condition_DET** могут использоваться в наборе **depends_on**.

F.3.7.3.3 Независимые типы элементов данных свойств (Non_dependent_P_DET)

Сущность **non_dependent_P_DET** — это свойства, не зависящие явно от некоторых условий.

Пример представления на языке EXPRESS:

```
*)
ENTITY non_dependent_P_DET
SUBTYPE OF (property_DET);
END_ENTITY; -- non_dependent_P_DET
(*
```

Примечание 1 — Три указанных подтипа (**condition_DET, dependent_P_DET и non_dependent_P_DET**) сущности **property_DET** используются для кодирования различных видов свойств (см. раздел 7). Сущность **condition_DET** используется для контекстных параметров. Сущность **dependent_P_DET** используется для контекстно-зависимых характеристик. Сущность **non_dependent_P_DET** используется для характеристики продукта.

Примечание 2 — Атрибут **depends_on** сущности **dependent_P_DET** используется для кодирования атрибута «Условие» для свойств.

F.3.7.4 Задание значений класса (Class_value_assignment)

Свойства, имеющие значение класса, — это свойства, значение которых не может быть назначено индивидуально для реализации класса. Оно может быть назначено только для всех реализаций, принадлежащих классу. Такие свойства объявляются путем их включения в список свойств подкласса **sub_class_properties** сущности **item_class**. Затем такому свойству может быть задано значение, верное для всех реализаций любого класса предметов **item_class,** являющегося подклассом для класса, где объявлено рассматриваемое свойство, имеющее значение класса, или в самом этом классе. Значение свойства, имеющего значение класса, задается для класса предметов **item_class** с помощью атрибута **class_value_assignment,** на который производится ссылка атрибутом **class_constant_value** данного класса.

Примечание — Свойства, имеющие значение класса, могут относиться к любому типу данных.

Пример представления на языке EXPRESS:

```
*)
ENTITY class_value_assignment;
    super_class_defined_property: property_BSU;
    assigned_value: primitive_value;
WHERE
    WR1: definition_available_implies (super_class_defined_property,
            compatible_data_type_and_value (super_class_defined_property.
            definition [1]\property_DET.domain, assigned_value));
END_ENTITY; -- class_value_assignment
(*
```

Определения атрибутов:

super_class_defined_property: ссылка на свойство (определенное в классе или в любом его суперклассе, как принадлежащее набору **sub_class_properties**), которому задается значение **assigned_value**.

assigned_value: значение, заданное свойству, корректному для целого класса, ссылающегося на рассматриваемую реализацию **class_value_assignment** набора **class_constant_value**, и для всех его подклассов.

Пояснения к тексту программы:

WR1: тип значения, заданного свойству, определенному в суперклассе **super_class_defined_properties,** должен быть совместим с областью значений свойств **super_class_defined_properties**.

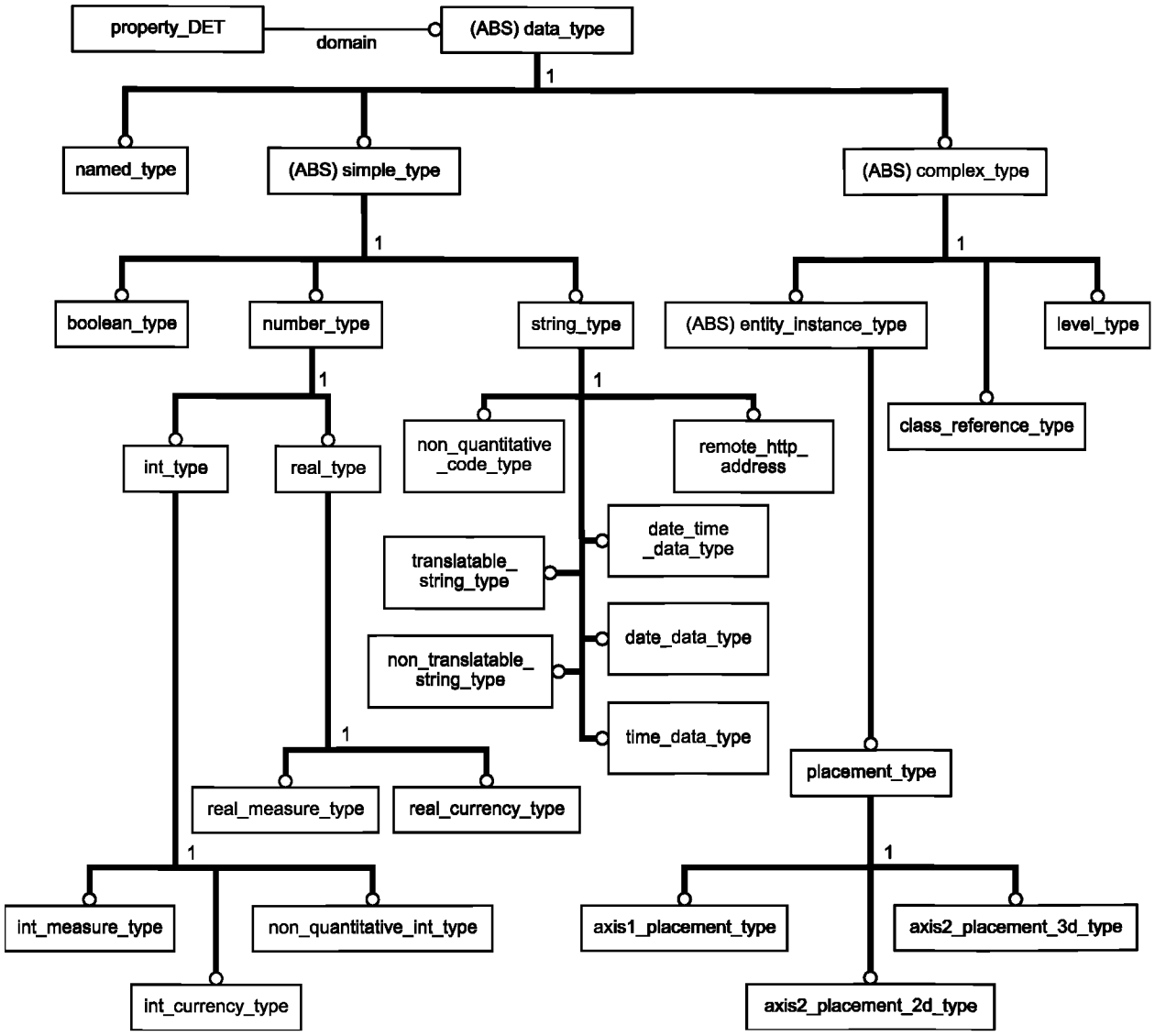


Рисунок F.11 — Иерархия сущностей для системы типов

property_DET	Тип элемента данных свойства
domain	Область
(ABS) data_type	Тип данных
named_type	Поименованный тип
(ABS) simple_type	Простой тип
(ABS) complex_type	Комплексный тип
boolean_type	Булев тип
number_type	Числовой тип
string_type	Строчный тип
(ABS) entity_instance_type	Тип реализации сущности
level_type	Тип уровня
class_reference_type	Тип ссылки на класс

int_type	Целый тип
real_type	Действительный тип
non_quantitative_code_type	Неколичественный кодовый тип
remote_http_address	Удаленный http-адрес
translatable_string_type	Тип переводимой строки
date_time_data_type	Тип данных о дате и времени
non_translatable_string_type	Непереводимый строчный тип
date_data_type	Тип данных о дате
time_data_type	Тип данных о времени
placement_type	Тип размещения
real_measure_type	Тип действительной меры
real_currency_type	Тип действительной валюты
int_measure_type	Тип целой меры
non_quantitative_int_type	Неколичественный целый тип
axis1_placement_type	Тип размещения по оси 1
axis2_placement_3d_type	3d-тип размещения по оси 2
int_currency_type	Тип целой валюты
axis2_placement_2d_type	2d-тип размещения по оси 2

F.3.8 Область данных: система типов

Данный раздел содержит определения представлений типов данных **property_DET**. На рисунке F.11 в виде модели планирования показана иерархия сущностей для типов данных.

F.3.8.1 Общие положения

В отличие от других словарных элементов (Поставщики, Классы, Свойства) идентификация с понятием базовой семантической единицы не является обязательной для типа данных **data_type,** так как она в большинстве случаев прикреплена прямо к типу данных **property_DET** и, таким образом, не требует идентификации. При этом сущности **data_type_BSU** и **data_type_element** могут использоваться для уникальной идентификации там, где это целесообразно. Это обеспечивает повторное использование рассматриваемого определения типа в другом определении **property_DET** даже вне текущего физического файла.

F.3.8.1.1 БСЕ типа данных (Data_type_BSU)

Сущность **data_type_BSU** позволяет идентифицировать элемент типа данных **data_type_element**.

Пример представления на языке EXPRESS:

```
*)
ENTITY data_type_BSU
SUBTYPE OF(basic_semantic_unit);
    SELF\basic_semantic_unit.code: data_type_code_type;
    name_scope: class_BSU;
DERIVE
    absolute_id: identifier :=
            name_scope.defined_by.absolute_id (* Supplier*)
            + sep_id + dic_identifier; (* Data_type *)
INVERSE
    defining_class: SET OF class FOR defined_types;
UNIQUE
    absolute_id;
WHERE
    WR1: is_subclass(defining_class [1], name_scope.definition [1]);
END_ENTITY; -- data_type_BSU
(*
```

Определения атрибутов:

code: дает уникальную идентификацию типа данных для всех онтологий, определенных рассматриваемым поставщиком **name_scope.defined_by**.

name_scope: ссылка на класс, на уровне которого (или ниже которого) рассматриваемый элемент типа данных доступен для ссылки атрибутом **defined_type**.

absolute_id: уникальная идентификация данного свойства.

defining_class: класс, объявляющий данный тип данных **data_type** доступным для использования в описании продукта.

Пояснения к тексту программы:

WR1: класс, используемый в атрибуте **name_scope,** является суперклассом для класса, где данный тип **data_type** определен.

Примечание — Атрибут **name_scope** используется для кодирования ссылки на класс, которому принадлежит соответственный тип данных. Данный атрибут, кроме сущности **data_type_element** (смотри ниже), является частью кода атрибута «Видимый тип» класса.

F.3.8.1.2 Элемент типа данных (Data_type_element)

Сущность **data_type_element** описывает типы словарных элементов. Отметим, что нет необходимости в каждом отдельном случае для заданного типа данных **data_type** иметь БСЕ и словарный элемент **dictionary_element,** потому что атрибут **property_DET** может ссылаться на тип данных **data_type** прямо. Использование БСЕ соотношений необходимо, если поставщик ссылается на один и тот же тип в различных физических файлах.

Пример представления на языке EXPRESS:

```
*)
ENTITY data_type_element
SUBTYPE OF(dictionary_element);
    SELF\dictionary_element.identified_by: data_type_BSU;
    names: item_names;
    type_definition: data_type;
END_ENTITY; -- data_type_element
(*
```

Определения атрибутов:

identified_by: БСЕ, задающий описанный элемент **data_type_element**.

names: имена, дающие описания определенного элемента **data_type_element.**

type_definition: описание типа элемента **data_type_element**.

Примечание — Повторно объявленный атрибут **identified_by** используется для кодирования ссылки на БСЕ, к которому относится данный элемент типа данных **data_type_element**. Данный элемент, кроме сущности **data_type_BSU** (смотри выше), используется для кодирования атрибута класса «Видимые типы».

F.3.8.2 Система типов

F.3.8.2.1 Тип данных (Data_type)

Сущность **data_type** служит общим супертипом для сущностей, используемых для указания типа ассоциированного типа элемента данных **DET**.

Пример представления на языке EXPRESS:

```
*)
ENTITY data_type
ABSTRACT SUPERTYPE OF(ONEOF(
    simple_type,
    complex_type,
    named_type));
    constraints: SET [0:?] OF domain_constraint;
WHERE
    WR1: QUERY (cons <* constraints
            |NOT correct_constraint_type (cons, SELF)) = [ ];
END_ENTITY; -- data_type
(*
```

Определения атрибутов:

constraints: набор ограничений, ограничивающих область значений типа данных.

Примечание — Каждое ограничение области в атрибуте **constraints** должно быть выполнено. Таким образом, атрибут **constraints** есть сопряжение ограничений.

Пояснение к тексту программы:

WR1: набор ограничений области должен определять ограничения, совместимые с областью значений типа данных.

F.3.8.2.2 Простой тип (Simple_type)

Сущность **simple_type** служит общим супертипом для сущностей, используемых для указания простого ассоциированного типа элемента данных **DET**.

Пример представления на языке EXPRESS:

```
*)
ENTITY simple_type
ABSTRACT SUPERTYPE OF(ONEOF(
    number_type,
    boolean_type,
    string_type))
SUBTYPE OF(data_type);
    value_format: OPTIONAL value_format_type;
END_ENTITY; -- simple_type
(*
```

Определения атрибутов:

value_format: вспомогательное кодирование формата значений свойств.

Примечание 1 — Атрибут **value_format** сущности **simple_type** используется для кодирования атрибута «Формат значения» свойства.

Примечание 2 — Если какое-либо ограничение **string_pattern_constraint** применяется к значению простого типа, то оно имеет преимущество перед форматом значения **value_format**.

F.3.8.2.3 Числовой тип (Number_type)

Сущность **number_type** задает значения элементов **DET** числового типа.

Пример представления на языке EXPRESS:

```
*)
ENTITY number_type
ABSTRACT SUPERTYPE OF(ONEOF(
    int_type,
    real_type,
    rational_type))
SUBTYPE OF(simple_type);
END_ENTITY; -- number_type
(*
```

F.3.8.2.4 Целый тип (Int_type)

Сущность **int_type** задает значения элементов **DET** типа **INTEGER**.

Пример представления на языке EXPRESS:

```
*)
ENTITY int_type
SUPERTYPE OF(ONEOF(
    int_measure_type,
    int_currency_type,
    non_quantitative_int_type))
SUBTYPE OF(number_type);
END_ENTITY; -- int_type
(*
```

F.3.8.2.5 Тип целой меры (Int_measure_type)

Сущность **int_measure_type** задает значения элементов **DET**, являющихся мерами целого типа. Она указывает единицу измерения или идентификатор единицы измерения **(unit_id),** выражающей значения, обмениваемые как отдельные целые. Она может также устанавливать альтернативные единицы измерения или идентификаторы альтернативных единиц измерения, допустимые для использования, когда каждое значение явно ассоциировано со своей единицей измерения.

Примечание 1 — Использование либо единицы **unit**, либо идентификатора единицы **unit_id** обязательно. В случае, когда присутствуют оба, **unit** имеет преимущество.

Примечание 2 — Если присутствуют и атрибуты **alternative_unit**, и атрибуты **alternative_unit_ids**, и они имеют одинаковый размер, то атрибут **alternative_unit** имеет преимущество.

Примечание 3 — Идентификатор словарной единицы **dic_unit_identifier**, используемый и в атрибуте **unit_id**, и в атрибуте **alternative_unit_ids** является идентификатором единицы измерения, который может быть разрешен в атрибуте **dic_unit** из сервера ИСО/ТС 29002-20.

Примечание 4 — Каждая словарная единица измерения **dic_unit**, определенная в атрибуте **alternative_unit**, и каждая словарная единица измерения **dic_unit**, идентифицированная в атрибуте **alternative_unit_ids**, должна быть ассоциирована со строчным представлением **string_representation**. Его текстовое представление **text_representation** может быть использовано для характеристики альтернативной единицы измерения, используемой на уровне реализации.

Пример представления на языке EXPRESS:

```
*)
ENTITY int_measure_type
SUBTYPE OF(int_type);
    unit: OPTIONAL dic_unit;
    alternative_units: OPTIONAL LIST [1:?] OF dic_unit;
    unit_id: OPTIONAL dic_unit_identifier;
    alternative_unit_ids: OPTIONAL LIST [1:?] OF dic_unit_identifier;
WHERE
    WR1: EXISTS(unit) OR EXISTS(unit_id);
    WR2: NOT EXISTS(alternative_units) OR
            NOT EXISTS(alternative_unit_ids) OR
            (SIZEOF(alternative_units) = SIZEOF(alternative_unit_ids));
    WR3: NOT EXISTS(alternative_units)
            OR (QUERY (un <* SELF.alternative_units
            |NOT EXISTS (un.string_representation))
            = [ ] );
END_ENTITY; -- int_measure_type
(*
```

Определения атрибутов:

unit: единица измерения по умолчанию, ассоциированная со значением целочисленной меры **int_measure_type**.

alternative_unit: список прочих единиц измерения, которые могут быть использованы для выражения значений целочисленной меры **int_measure_type**.

Примечание 5 — Списочный порядок гарантирует, что атрибуты **alternative_unit** и **alternative_unit_id**, если оба существуют, определяют ту же единицу в том же порядке.

unit_id: идентификатор единицы измерения по умолчанию, ассоциированный с описанной мерой.

Примечание 6 — Атрибут **unit** и атрибут **unit_id** используются для кодирования атрибута «Единица измерения» свойства. Если имеются оба атрибута, то unit имеет преимущество

Примечание 7 — Если значение свойства с областью **int_measure_type** обменивается как отдельное целое, то данное значение выражается с помощью атрибутов **unit** или **unit_id**.

alternative_unit_ids: список идентификаторов прочих единиц измерения, используемых для выражения значения в целочисленной мере **int_measure_type**.

Примечание 8 — Если значение свойства с областью **int_measure_type** оценивается либо атрибутом **unit**, либо атрибутом **alternative_unit**, либо идентифицируется атрибутом **alternative_unit_id**, то это значение не может быть представлено как отдельное целое. Оно должно быть представлено парой «значение, единица измерения».

Пояснения к тексту программы:

WR1: должен существовать хотя бы один из двух атрибутов **unit** и **unit_id**.

WR2: если существуют и атрибут **alternative_unit,** и атрибут **alternative_unit_id**, то они должны иметь одинаковую длину.

WR3: каждая словарная единица измерения **dic_unit,** представленная в альтернативных единицах измерения **alternative_unit,** должна иметь строчное представление **string_representation.**

Дополнительные пояснения:

IP1: идентификатор **dic_unit_identifier,** используемый и в атрибуте **unit_id,** и в атрибуте **alternative_unit_ids,** должен быть разрешен в **dic_unit** из существующего сервера ISO/TS 29002-20.

IP2: если имеются и атрибут **unit,** и атрибут **unit_id,** то они должны определять одну и ту же единицу измерения.

IP3: если имеются и атрибут **alternative_unit,** и атрибут **alternative_unit_ids**, то они должны определять тот же список единиц измерения в том же порядке.

IP4: если атрибут **alternative_unit_ids** существует, то все единицы, которые данный атрибут идентифицирует, должны быть разрешены в словарную единицу **dic_unit,** имеющую строчное представление **string_representation.**

F.3.8.2.6 Целый валютный тип (Int_currency_type)

Сущность **int_currency_type** задает значение элементам **DET**, которые являются целыми валютами.

Пример представления на языке EXPRESS:

```
*)
ENTITY int_currency_type
SUBTYPE OF(int_type);
    currency: OPTIONAL currency_code;
END_ENTITY; -- int_currency_type
(*
```

Определения атрибутов:

currency: ассоциированный код валюты, описанной в соответствии с ИСО 4217. Если он отсутствует, то валютный код нужно обменивать вместе с соответствующими данными (значениями).

F.3.8.2.7 Неколичественный целый тип (Non_quantitative_int_type)

Сущность **non_quantitative_int_type** — это тип нумерации, где элементы нумерации представлены целым значением (см. также сущность неколичественный кодовый тип **non_quantitative_code_type** и рисунок 12).

Пример представления на языке EXPRESS:

```
*)
ENTITY non_quantitative_int_type
SUBTYPE OF(int_type);
    domain: value_domain;
WHERE
    WR1: QUERY(v <* domain.its_values |
            'ISO13584_IEC61360_DICTIONARY_SCHEMA.VALUE_CODE_TYPE' IN
            TYPEOF(v.value_code)) = [ ];
END_ENTITY; -- non_quantitative_int_type
(*
```

Определения атрибутов:

domain: множество перенумерованных значений, описанных в сущности **value_domain**.

Пояснения к тексту программы:

WR1: значения, ассоциированные со списком **domain.its_value,** не должны содержать тип **value_code_type.**

F.3.8.2.8 Действительный тип (Real_type)

Сущность **real_type** задает значения элементов **DET** типа **REAL**.

Пример представления на языке EXPRESS:

```
*)
ENTITY real_type
SUPERTYPE OF(ONEOF(
    real_measure_type,
    real_currency_type))
SUBTYPE OF(number_type);
END_ENTITY; -- real_type
(*
```

F.3.8.2.9 Тип действительной меры (Real_measure_type)

Сущность **real_measure_type** задает значения элементам **DET** с мерой типа **REAL**. Она указывает единицы измерения или идентификаторы единиц измерения, в которых выражаются значения, обмениваемые как отдельные действительные величины. Данная сущность может также устанавливать альтернативные единицы измерения или идентификаторы альтернативных единиц измерения, допустимые для использования, когда каждое значение явно ассоциировано со своей единицей измерения.

Примечание 1 — Наличие либо атрибута **unit**, либо атрибута **unit_id** обязательно. Если они имеются оба, то атрибут **unit** имеет преимущество.

Примечание 2 — Если имеются и атрибут **alternative_unit**, и атрибут **alternative_unit_ids**, то они имеют одинаковый размер, а атрибут **alternative_unit** имеет преимущество.

Примечание 3 — Идентификаторы **dic_unit_identifier**, используемые и в атрибуте **unit_id**, и в атрибуте **alternative_unit_id**, являются идентификаторами единиц измерения, которые могут быть разрешены в словарной единице **dic_unit** из сервера ИСО/ТС 29002-20.

Примечание 4 — Каждая словарная единица **dic_unit**, определенная в атрибуте **alternative_unit**, и каждая словарная единица **dic_unit**, идентифицированная в атрибуте **alternative_unit_ids**, должна быть ассоциирована со строчным представлением **string_representation**, чье текстовое представление **text_representation** может быть использовано для характеристики альтернативной единицы измерения, используемой на уровне реализации.

Пример представления на языке EXPRESS:

```
*)
ENTITY real_measure_type
SUBTYPE OF(real_type);
    unit: OPTIONAL dic_unit;
    alternative_units: OPTIONAL LIST [1:?] OF dic_unit;
    unit_id: OPTIONAL dic_unit_identifier;
    alternative_unit_ids: OPTIONAL LIST [1:?] OF dic_unit_identifier;
WHERE
    WR1: EXISTS(unit) OR EXISTS(unit_id);
    WR2: NOT EXISTS(alternative_units)
            OR NOT EXISTS(alternative_unit_ids)
```

```
                OR (SIZEOF(alternative_units) =
                SIZEOF(alternative_unit_ids));
        WR3: NOT EXISTS(alternative_units)
                OR (QUERY (un <* SELF.alternative_units
                |NOT EXISTS (un.string_representation))
                = [ ] );
    END_ENTITY; -- real_measure_type
    (*
```

Определения атрибутов:

unit: единица измерения по умолчанию, ассоциированная со значением **real_measure_type.**

alternative_unit: список прочих единиц измерения, которые могут использоваться для выражения значений **real_measure_type.**

Примечание 5 — Используемый списочный порядок гарантирует, что и атрибут **alternative_unit**, и атрибут **alternative_unit_ids**, если они существуют, определяют ту же единицу в том же порядке.

unit_id: идентификатор единицы измерения по умолчанию, ассоциированный с описанной мерой.

Примечание 6 — И атрибут **unit**, и атрибут **unit_id** используются для кодирования атрибута «Единица измерения». Если они имеются оба, то атрибут **unit** имеет преимущество.

Примечание 7 — Если значение свойства с областью **real_measure_type** обменивается как отдельная действительная величина, то данное значение выражается в единицах меры **unit** или **unit_id**.

alternative_unit_ids: список идентификаторов прочих единиц измерения, используемых для выражения значений **real_measure_type.**

Примечание 8 — Если значение свойства с областью **real_measure_type** оценивается либо с помощью атрибута **unit**, либо с помощью атрибута **alternative_unit**, либо идентифицируется атрибутом **alternative_unit_ids**, то его значение не может быть представлено как отдельное действительное значение. Оно может быть представлено парой «значение, единица измерения».

Пояснения к тексту программы:

WR1: должен существовать хотя бы один из двух атрибутов **unit** или **unit_id**.

WR2: если и атрибут **alternative_unit,** и атрибут **alternative_unit_ids** существуют, то они должны иметь одинаковую длину.

WR3: каждая словарная единица **dic_unit** с атрибутом **alternative_unit** должна иметь строчное представление **string_representation.**

Дополнительные пояснения:

IP1: идентификаторы **dic_unit_identifier,** используемые и в идентификаторе единицы измерения **unit_id,** и в альтернативных идентификаторах **alternative_unit_id,** должны быть разрешимы в словарных единицах измерения **dic_unit** из существующего сервера ИСО/ТС 29002-20.

IP2: если существуют и атрибут **unit,** и атрибут **unit_id**, то они должны определять одинаковые единицы измерения.

IP3: если существуют и атрибуты **alternative_unit**, и атрибуты **alternative_unit_ids**, то они должны определять тот же список единиц измерения в том же порядке.

IP4: если существует атрибут **alternative_unit_ids**, то все единицы измерения, которые данный атрибут задает, должны разрешаться в словарной единице измерения **dic_unit,** имеющей строчное представление **string_representation**.

F.3.8.2.10 Действительный валютный тип (Real_currency_type)

Сущность **real_currency_type** определяет действительные валюты.

Пример представления на языке EXPRESS:

```
    *)
    ENTITY real_currency_type
    SUBTYPE OF(real_type);
        currency: OPTIONAL currency_code;
    END_ENTITY; -- real_currency_type
    (*
```

Определения атрибутов:

currency: ассоциированный код описанной валюты в соответствии с ИСО 4217. Если данный код отсутствует, то валютный код обменивается вместе с соответствующими данными (значениями).

F.3.8.2.11 Рациональный тип (Rational_type)

Сущность **rational_type** задает значения элементов **DET** с типом **rational**.

Примечание — В ИСО 13584-32 рациональные значения представлены тремя целыми элементами расширяемого языка разметки XML: целая часть, числитель и знаменатель. В ИСО 13584-24:2003 рациональные значения представлены массивом из трех целых: целая часть, числитель и знаменатель.

Пример представления на языке EXPRESS:

```
*)
ENTITY rational_type
SUPERTYPE OF(
    rational_measure_type)
SUBTYPE OF(number_type);
END_ENTITY; -- rational_type
(*
```

F.3.8.2.12 Тип рациональной меры (Rational_measure_type)

Сущность **rational_measure_type** задает значения элементов **DET** с типом меры **RATIONAL.**

Пример — Диаметр винта 4 1/8 дюйма.

Сущность **rational_measure_type** указывает единицу измерения или идентификатор единицы измерения, выражающей обмениваемые значения как рациональные. Данная сущность может также устанавливать альтернативные единицы измерения, а также идентификаторы альтернативной единицы измерения, допустимые для использования, когда каждое значение явно ассоциировано со своей единицей измерения.

Примечание 1 — Присутствие либо атрибута **unit**, либо **unit_id** обязательно. Если присутствуют оба атрибута, то атрибут **unit** имеет преимущество.

Примечание 2 — Если присутствуют и атрибут **alternative_unit**, и атрибут **alternative_unit_ids**, то они имеют одинаковый размер, и атрибут **alternative_unit** имеет преимущество.

Примечание 3 — Идентификаторы словарной единицы измерения dic_**unit_id**entifier, используемые и в атрибуте **unit_id**, и в атрибуте **alternative_unit_ids**, — это идентификаторы единицы измерения, разрешаемые в **dic_unit** из сервера ИСО/ТС 29002-20.

Примечание 4 — Каждая единица измерения **dic_unit**, определенная в атрибуте **alternative_unit**, и каждая единица измерения **dic_unit**, идентифицированная в атрибуте **alternative_unit_ids**, ассоциирована со строчным представлением **string_representation**. Ее текстовое представление **text_representation** может быть использовано для характеристики альтернативной единицы измерения, используемой на уровне реализации.

Пример представления на языке EXPRESS:

```
*)
ENTITY rational_measure_type
SUBTYPE OF(rational_type);
    unit: OPTIONAL dic_unit;
    alternative_units: OPTIONAL LIST [1:?] OF dic_unit;
    unit_id: OPTIONAL dic_unit_identifier;
    alternative_unit_ids: OPTIONAL LIST [1:?] OF dic_unit_identifier;
WHERE
    WR1: EXISTS(unit) OR EXISTS(unit_id);
    WR2: NOT EXISTS(alternative_units)
            OR NOT EXISTS(alternative_unit_ids)
            OR (SIZEOF(alternative_units) =
            SIZEOF(alternative_unit_ids));
    WR3: NOT EXISTS(alternative_units) OR (QUERY (un <*
            SELF.alternative_units |
            NOT EXISTS (un.string_representation)) = [ ]);
END_ENTITY; -- rational_measure_type
(*
```

Определения атрибутов:

unit: единица измерения по умолчанию, ассоциированная со значением типа рациональной меры **rational_measure_type.**

alternative_unit: список прочих единиц измерения, которые могут быть использованы для выражения значений типа рациональной меры **rational_measure_type.**

Примечание 5 — Данный списочный порядок гарантирует, что если существуют и атрибут **alternative_unit**, и атрибут **alternative_unit_ids**, то определяются те же единицы измерения и в том же порядке.

unit_id: идентификатор единиц измерения по умолчанию, ассоциированный с описанной мерой.

Примечание 6 — И атрибут **unit**, и атрибут **unit_id** используются для кодирования атрибута «Единица измерения» свойств. Если существуют оба, то атрибут **unit** имеет преимущество.

Примечание 7 — Если значение свойства с областью **rational_measure_type** обменивается как отдельное рациональное число, то данное значение выражается в единицах измерения **unit** или **unit_id**.

alternative_unit_ids: список идентификаторов прочих единиц измерения, используемых для выражения значений типа рациональной меры **rational_measure_type.**

Примечание 8 — Если значение свойства с областью **rational_measure_type** оценивается в единицах измерения, либо определенных атрибутом **alternative_unit**, либо идентифицированных атрибутом **alternative_unit_ids**, то это значение не может быть представлено как отдельное рациональное число. Оно должно быть представлено парой «значение, единица измерения».

Пояснения к тексту программы:

WR1: должен существовать хотя бы один из атрибутов **unit** или **unit_id**.

WR2: если существуют и атрибут **alternative_unit,** и атрибут **alternative_unit_ids**, то они должны иметь одинаковую длину.

WR3: каждая словарная единица измерения **dic_unit** в атрибуте **alternative_unit** должна иметь строчное представление **string_representation.**

Дополнительные пояснения:

IP1: идентификатор **dic_unit_identifier**, используемый и в атрибуте **unit_id,** и в атрибуте **alternative_unit_ids,** должен разрешаться в **dic_unit** из существующего сервера ИСО/ТС 29002-20.

IP2: если существует и атрибут **unit,** и атрибут **unit_id**, то они должны определять одинаковые единицы измерения.

IP3: если существуют и атрибут **alternative_unit,** и атрибут **alternative_unit_ids**, то они должны определять тот же список единиц измерения в том же порядке.

IP4: если существует атрибут **alternative_unit_ids**, то все единицы измерения, которые идентифицирует данный атрибут, должны разрешаться в **dic_unit,** имеющей строчное представление **string_representation**.

F.3.8.2.13 Булевский тип (boolean_type)

Сущность **boolean_type** задает значения элементов **DET** булевского типа.

Пример представления на языке EXPRESS:

```
*)
ENTITY boolean_type
SUBTYPE OF(simple_type);
END_ENTITY; -- boolean_type
(*
```

F.3.8.2.14 Строчный тип (String_type)

Сущность **string_type** задает значения элементов **DET** строчного типа.

Пример представления на языке EXPRESS:

```
*)
ENTITY string_type
SUPERTYPE OF (ONEOF (
    translatable_string_type,
    non_translatable_string_type,
    URI_type,
    non_quantitative_code_type,
    date_time_data_type,
    date_data_type,
    time_data_type))
SUBTYPE OF(simple_type);
END_ENTITY; -- string_type
(*
```

F.3.8.2.15 Переводимый строчный тип (Translatable_string_type)

Сущность **translatable_string_type** задает значения элементов **DET** строчного типа, которые могут быть представлены различными строками на различных языках.

Примечание 1 — Значения таких свойств не могут быть использованы для идентификации продуктов.

Примечание 2 — Значения таких свойств могут быть либо простыми строчными значениями **string_value**, если глобальное назначение **global_language_assignment** определяет текущий язык, либо переведенными строчными значениями **translated_string_value**, где каждое значение строки ассоциировано с языком.

Примечание 3 — Два значения одного свойства с типом данных **data_type**, равным **translatable_string_type**, можно сравнивать на предмет равенства, только если такое соответствующее свойство, как исходный язык **source_language**, определено как часть его административных данных **administrative_data**, и если указанные значения доступны в данном исходном языке **source_language**. При этом не предполагается, что на языке, отличном от исходного языка **source_language**, тот же смысл представлен той же строкой.

Пример представления на языке EXPRESS:

```
*)
ENTITY translatable_string_type
SUBTYPE OF(string_type);
END_ENTITY; -- translatable_string_type
(*
```

F.3.8.2.16 Непереводимый строчный тип (Non_**translatable_string_type**)

Сущность **non_translatable_string_type** задает значения элементов **DET** строчного типа, представленных неизменным способом на любом языке.

Примечание — Значения таких свойств могут использоваться для идентификации продуктов.

Пример представления на языке EXPRESS:

```
*)
ENTITY non_translatable_string_type
SUBTYPE OF(string_type);
END_ENTITY; -- non_translatable_string_type
(*
```

F.3.8.2.17 Тип универсального идентификатора ресурсов (URI_type)

Сущность **URI_type** задает значения элементов **DET** строчного типа, содержащих универсальный идентификатор ресурсов **URI**.

Примечание — Сущность **URI_type** прежде всего идентифицирует универсальный идентификатор ресурсов **URL**.

Пример представления на языке EXPRESS:

```
*)
ENTITY URI_type
SUBTYPE OF(string_type);
END_ENTITY; -- URI_type
(*
```

F.3.8.2.18 Date_time_data_type

Сущность **date_time_data_type** задает значения элементов DET строчного типа. Она содержит особый момент времени, указанный в соответствии с конкретным представлением по ИСО 8601.

Примечание 1 — Только подмножество лексических представлений, допускаемое ИСО 8601, можно использовать для задания значений **date_time_data_type**. См. пояснение IP1 ниже.

Примечание 2 — Вышеуказанные ограничения ИСО 8601 на рассматриваемое представление соответствуют требованиям Схемы XML.

Пример представления на языке EXPRESS:

```
*)
ENTITY date_time_data_type
SUBTYPE OF(string_type);
END_ENTITY; -- date_time_data_type
(*
```

Дополнительные пояснения:

IP1: значение свойства с типом данных **date_time_data_type** должно соответствовать нижеследующему лексическому представлению, являющемуся подмножеством лексического представления, определенного ИСО 8601. Данное лексическое представление — это определенный ИСО 8601 расширенный формат CCYY-MM—DDThh:mm:ss, где «CC» представляет век (первому веку соответствует «00»), «YY» — год, «MM» — месяц и «DD» — день. Вспомогательный знак «-» (минус) спереди указывает отрицательное число. Если минус опущен, то предполагается «+» (плюс). Буква «T» — это разделитель даты и времени. Обозначения «hh», «mm», «ss» — это час, минута и секунда соответственно. Дополнительные цифры могут использоваться для указания долей секунды при необходимости. При этом используется формат ss.ss... с любым количеством цифр после десятичной точки. Дробная часть секунды указывается по особому требованию. Прочие части рассматриваемой лексической формы присутствуют всегда. Чтобы указать значение года, превышающее 9999, нужны дополнительные цифры слева. Спереди ставят нули, если значение года требует менее четырех цифр. В противном случае нули не используются. Например, год 0000 запрещен. Поле CCYY должно иметь, по крайней мере, четыре цифры. Поля MM, DD, SS, hh, mm и ss представляются двумя цифрами каждое (не считая долей секунды). Предшествующие нули используются, если в рассматриваемом поле не хватает значащих цифр. За указанным представлением может сразу идти символ «Z», указывающий Координированное Универсальное Время (UTC). Для указания временного пояса, т.е. сдвига между местным временем и Координированным Универсальным Временем, используется знак «+» или «-». Далее указывается временной сдвиг (по отношению к UTC) в формате hh:mm (Примечание: указание минут обязательно). В ИСО 8601 даны требования к записи значений в различных полях. Если указывается временной пояс, то наличие и часов, и минут обязательно.

Пример — Чтобы указать время 1:20 после полудня 31 мая 1999 года по восточному календарю, отстающее на 5 часов от Координированного Универсального Времени (UTC), нужно записать: 1999-05-31T13:20:00—05:00.

F.3.8.2.19 Тип данных о дате (Date_**data_type**)

Сущность **date_data_type** задает значения элементов **DET** строчного типа. Она содержит особую календарную дату, установленную в соответствии с конкретным представлением по ИСО 8601.

Примечание 1 — Только подмножество лексических представлений, соответствующих ИСО 8601, может быть использовано как значение сущности **date_data_type**. См. пояснение IP1 ниже.

Примечание 2 — Вышеуказанное ограничение ИСО 8601 на представления совпадает с ограничением, установленным Схемой XML.

Пример представления на языке EXPRESS:

```
*)
ENTITY date_data_type
SUBTYPE OF(string_type);
END_ENTITY; -- date_data_type
(*
```

Дополнительные пояснения:

IP1: значение свойства с типом данных **date_data_type** должно соответствовать нижеследующему лексическому представлению, являющемуся подмножеством лексических представлений, установленных ИСО 8601. Лексическое представление для типа данных **date_data_type** является сокращенным (усеченным справа) лексическим представлением для типа данных **date_time_data_type:** CCYY-MM-DF. Усечение слева недопустимо. Для типа данных **date_time_data_type** может быть использован вспомогательный идентификатор часового пояса. Если значение года выходит из интервала 0001—9999, то в левой части данного представления добавляются цифры и знак «-».

Пример — Представление 1999-05-31 — это представление типа date_data_type для 31 мая 1999 года.

F.3.8.2.20 Тип времени суток (Time_**data_type**)

Сущность **time_data_type** задает значения элементов **DET** строчного типа. Она содержит указание особого момента времени в соответствии с требованиями ИСО 8601. Это значение **time_data_type** дает момент времени, повторяющийся каждый день.

Примечание 1 — Только подмножество лексических представлений, определенных ИСО 8601, может использоваться для значения **time_data_type**. См. пояснение IP1 ниже.

Примечание 2 — Вышеуказанное ограничение ИСО 8601 совпадает с ограничением, установленным Схемой XML.

Примечание 3 — Рассматриваемое лексическое представление допускает использование вспомогательного идентификатора часового пояса. Поэтому значения **time_data_type** частично упорядочены, чтобы исключить возможность произвольно задавать порядок следования двух значений, одно из которых имеет часовой пояс, а другое — нет.

Пример представления на языке EXPRESS:

```
*)
ENTITY time_data_type
SUBTYPE OF(string_type);
END_ENTITY; -- time_data_type
(*
```

Дополнительные пояснения:

IP1: значение свойства с типом данных **time_data_type** должно соответствовать нижеследующему лексическому представлению, являющемуся подмножеством лексических представлений, определенных ИСО 8601. Лексическое представление типа данных **time_data_type** — это усеченное слева лексическое представление типа **date_time_data_type** hh:mm:ss.sss со вспомогательным (по выбору) идентификатором часового пояса.

Пример — Представление 13:20:00—05:00 — это представление суточного времени time_data_type, соответствующее 1.20 после полудня для восточного стандартного времени, которое отстает на 5 часов от Координированного Универсального Времени (UTC).

F.3.8.2.21 Неколичественный кодовый тип (Non_quantitative_code_type)

Сущность **non_quantitative_code_type** — это тип нумерации, когда элементы нумерации представлены значением строки (смотри также сущность неколичественный целый тип **non_quantitative_int_type** и рисунок F.12).

Пример представления на языке EXPRESS:

```
*)
ENTITY non_quantitative_code_type
SUBTYPE OF(string_type);
    domain: value_domain;
```

```
WHERE
    WR1: QUERY(v <* domain.its_values |
    NOT('ISO13584_IEC61360_DICTIONARY_SCHEMA.VALUE_CODE_TYPE' IN
    TYPEOF(v.value_code))) = [ ];
END_ENTITY; -- non_quantitative_code_type
(*
```

Определения атрибутов:

domain: набор перенумерованных значений, описанных сущностью **value_domain**.

Пояснения к тексту программы:

WR1: значения, ассоциированные со списком **domain.its_value,** должны содержать только элементы типа **value_code_type.**

F.3.8.2.22 Комплексный тип (Complex_type)

Сущность **complex_type** дает определения типов, значения которых представлены как реализации языка EXPRESS.

Пример представления на языке EXPRESS:

```
*)
ENTITY complex_type
ABSTRACT SUPERTYPE OF(ONEOF(
    level_type,
    class_reference_type,
    entity_instance_type))
SUBTYPE OF(data_type);
END_ENTITY; -- complex_type
(*
```

F.3.8.2.23 Тип уровня (Level_type)

Тип данных **level_type** — это комплексный тип, указывающий, что значение свойства содержит от одного до четырех действительных или целых значений. Каждое из них описывается конкретным индикатором, указывающим смысл данного значения.

Примечание 1 — Значения реализаций **level_type** содержат значения только индикаторов, установленных атрибутом **levels**. Если некоторые из указанных значений недоступны, то они представляются нулями **null_value**.

Пример — Если атрибут level_type указывает, что только минимальные и типовые значения могут быть целыми, то рассматриваемая реализация содержит целые значения (или нулевые значения null_value) только для минимальных и типовых значений реализаций level_type.

Пример представления на языке EXPRESS:

```
*)
ENTITY level_type
SUBTYPE OF(complex_type);
    levels: LIST [1:4] OF UNIQUE level;
    value_type: simple_type;
WHERE
    WR1: ('ISO13584_IEC61360_DICTIONARY_SCHEMA.INT_MEASURE_TYPE'
            IN TYPEOF(value_type))
            OR ('ISO13584_IEC61360_DICTIONARY_SCHEMA.Real_measure_type'
            IN TYPEOF(value_type));
    WR2: NOT EXISTS(SELF.levels[2]) OR
            (SELF.levels[1] < SELF.levels[2]);
    WR3: NOT EXISTS(SELF.levels[2]) OR NOT EXISTS(SELF.levels[3]) OR
            (SELF.levels[2] < SELF.levels[3]);
    WR4: NOT EXISTS(SELF.levels[3]) OR NOT EXISTS(SELF.levels[4]) OR
            (SELF.levels[3] < SELF.levels[4]);
END_ENTITY; -- level_type
(*
```

Определения атрибутов:

уров: список уникальных индикаторов, указывающих, какие из заданных значений должны быть ассоциированы со свойством.

value_type: тип данных указанных значений атрибута **level_type.**

Пояснения к тексту программы:

WR1: тип собственного значения **SELF.value_type** — это либо тип целой меры **int_measure_type,** либо тип действительной меры **real_measure_type.**

WR2: порядок первого и второго уровней **level,** если оба существуют, должны соответствовать порядку нумерации типа **level**.

WR3: порядок второго и третьего уровней **level,** если оба существуют, должны соответствовать порядку нумерации типа **level**.

WR4: порядок третьего и четвертого уровней **level,** если оба существуют, должны соответствовать порядку нумерации типа **level**.

F.3.8.2.24 Уровень (level)

Сущность **level** определяет индикаторы, используемые для задания значений атрибута **level_type.**

Используемые индикаторы:

- **minimum**: наименьшее значение рассматриваемой величины, установленное для заданного набора рабочих условий, в которых компонент, устройство или оборудование является работоспособным и функционирует в соответствии с установленными требованиями;
- **nominal:** значение величины, используемое для обозначения и идентификации компонента, устройства, оборудования или системы;
- **typical**: обычно встречающееся значение величины, используемое для задания целей, установленных для указанного набора условий эксплуатации компонента, устройства, оборудования или системы;
- **maximum**: наибольшее значение величины, установленное для заданного набора условий эксплуатации, в которых компонент, устройство или оборудование является работоспособным и функционирует в соответствии с установленными требованиями.

Примечание 1 — Номинальное значение обычно округляют.

Пример — Автомобильный аккумулятор с номинальным напряжением 12 В имеет 6 элементов с типовым значением напряжения 2,2 В каждый. Типовое напряжение этой батареи равно 13,5 V. Напряжение может достигать максимального значения 14,5 В. Аккумулятор считается полностью разряженным, если напряжение падает ниже минимума 12,5 В.

Примечание 2 — Значения, установленные для определенного уровня свойства, определены в словаре.

Примечание 3 — Рекомендуется ограничивать тип уровня теми типами элементов данных **DET**, для которых множественность значений одной и той же характеристики является востребованным и общепринятым делом (например, для электронных компонентов промышленных установок).

Пример представления на языке EXPRESS:

```
*)
TYPE level = ENUMERATION OF(
    min, (* the minimal value of the physical quantity *)
    nom, (* the nominal value of the physical quantity *)
    typ, (* the typical value of the physical quantity *)
    max); (* the maximal value of the physical quantity *)
END_TYPE; -- level
(*
```

F.3.8.2.25 Тип ссылки на класс (Class_reference_type)

Сущность **class_reference_type** задает значения элементов **DET**, являющиеся реализациями класса. Она используется, как правило, для описания сборок или материалов (деталей), из которых состоит компонент.

Пример представления на языке EXPRESS:

```
*)
ENTITY class_reference_type
SUBTYPE OF(complex_type);
    domain: class_BSU;
END_ENTITY; -- class_reference_type
(*
```

Определения атрибутов:

domain: БСЕ класса **class_BSU,** ссылающаяся на класс, представляющий описанный тип.

F.3.8.2.26 Тип реализации сущности (Entity_instance_type)

Сущность **entity_instance_type** задает значения элементов **DET**, представляемых реализациями некоторых типов данных сущностей языка EXPRESS. Атрибут имени типа **type_names** позволяет выбрать допустимый тип данных. Указанные типы данных — это строки, образующие набор. Данный атрибут, вместе с функцией типа TYPEOF языка EXPRESS, примененный к некоторому значению, позволяет выполнить строгую проверку типа и выявить полиморфизм. Данная сущность описывается как подтип ниже для некоторых типов данных, допустимых для использования в словарных схемах.

Примечание — Если некоторая сущность языка EXPRESS — это значение некоторого элемента **DET**, тип данных которого — это тип реализации сущности **entity_instance_type,** то возможно скорректировать тип, применив функцию типа TYPEOF языка EXPRESS для данного значения **DET** и сравнив результаты данного действия со строками, содержащимися в атрибуте набора имен типов **type_names**.

Пример представления на языке EXPRESS:

```
*)
ENTITY entity_instance_type
SUBTYPE OF(complex_type);
    type_name: SET OF STRING;
END_ENTITY; -- entity_instance_type
(*
```

Определения атрибутов:

type_names: набор строк, дающих описание (в формате функции типа TYPEOF языка EXPRESS) имен типов данных сущности языка EXPRESS, принадлежащий результату применения функции типа TYPEOF языка EXPRESS, примененной к значению, ссылающемуся на рассматриваемую сущность, как на ее тип данных.

F.3.8.2.27 Тип размещения (Placement_type)

Сущность **placement_type** используется для значений элементов **DET**, являющихся реализациями типа данных сущности **placement**.

Примечание 1 — Сущности размещения **placement** импортируются из ИСО 10303-42. В соответствии с ИСО 10303-42, реализация **placement** может существовать, только если она относится к реализации контекста геометрического представления **geometric_representation_context** (в некоторой реализации представления). Следовательно, если некоторые свойства класса имеют реализации сущности **placement** как их значения, то данный класс содержит контекст геометрического представления **geometric_representation_context** (который определяет контекст указанных размещений) и само представление (которое соединяет указанные размещения placements с их контекстом). Ни **geometric_representation_context**, ни **representation** не импортируются в настоящий стандарт. Сущности **placement** не могут использоваться, если используются только схемы ИСО 13584-42. Указанные сущности вводятся как ресурсы для других частей ИСО 13584.

Примечание 2 — Сущности **Placement**, в частности, используются в ИСО 13584-32 (OntoML) и в ИСО 13584-25.

Пример представления на языке EXPRESS:

```
*)
ENTITY placement_type
SUPERTYPE OF(ONEOF(
    axis1_placement_type,
    axis2_placement_2d_type,
    axis2_placement_3d_type))
SUBTYPE OF(entity_instance_type);
WHERE
    WR1: 'GEOMETRY_SCHEMA.PLACEMENT'
            IN SELF\entity_instance_type.type_name;
END_ENTITY; -- placement_type
(*
```

Пояснения к тексту программы:

WR1: строка **'GEOMETRY_schema.PLACEMENT'** должна содержаться в наборе, определенном атрибутом **SELF\entity_instance_type.type_names**.

F.3.8.2.28 Тип размещения на оси 1(Axis1_placement_type)

Сущность **axis1_placement_type** задает значения элементов **DET**, являющихся реализациями типа данных сущности **axis1_placement** (см. ИСО 10303-42).

Пример представления на языке EXPRESS:

```
*)
ENTITY axis1_placement_type
SUBTYPE OF(placement_type);
WHERE
    WR1: 'GEOMETRY_SCHEMA.AXIS1_PLACEMENT' IN
            SELF\entity_instance_type.type_name;
END_ENTITY; -- axis1_placement_type
(*
```

Пояснения к тексту программы:

WR1: строка **'GEOMETRY_schema.AXIS1_PLACEMENT'** должна содержаться в наборе, определенном атрибутом **SELF\entity_instance_type.type_names**.

F.3.8.2.29 Тип 2d-размещения на оси 2 (Axis2_placement_2d_type)

Сущность **axis2_placement_2d_type** задает значения элементов **DET**, являющихся реализациями типа данных сущности **axis2_placement_2d** (см. ИСО 10303-42).

Пример представления на языке EXPRESS:

```
*)
ENTITY axis2_placement_2d_type
SUBTYPE OF(placement_type);
WHERE
    WR1: 'GEOMETRY_SCHEMA.AXIS2_PLACEMENT_2D'
            IN SELF\entity_instance_type.type_name;
END_ENTITY; -- axis2_placement_2d_type
(*
```

Пояснения к тексту программы:

WR1: строка **'GEOMETRY_schema.AXIS2_PLACEMENT_2D'** должна содержаться в наборе, определенном для атрибута **SELF\entity_instance_type.type_names**.

F.3.8.2.30 Тип 3d-размещения на оси 2 (Axis2_placement_3d_type)

Сущность **axis2_placement_3d_type** задает значения элементов **DET**, являющихся реализациями типа данных сущности **axis2_placement_3d** (см. ИСО 10303-42).

Пример представления на языке EXPRESS:

```
*)
ENTITY axis2_placement_3d_type
SUBTYPE OF(placement_type);
WHERE
    WR1: 'GEOMETRY_SCHEMA.AXIS2_PLACEMENT_3D'
            IN SELF\entity_instance_type.type_name;
END_ENTITY; -- axis2_placement_3d_type
(*
```

Пояснения к тексту программы:

WR1: строка **'GEOMETRY_schema.AXIS2_PLACEMENT_3D'** должна содержаться в наборе, определенном для атрибута **SELF\entity_instance_type.type_names**.

F.3.8.2.31 Поименованный тип (Named_type)

Сущность **named_type** позволяет ссылаться на другие типы с помощью БСЕ механизма.

Пример представления на языке EXPRESS:

```
*)
ENTITY named_type
SUBTYPE OF(data_type);
    referred_type: data_type_BSU;
END_ENTITY; -- named_type
(*
```

Определения атрибутов:

referred_type: БСЕ, идентифицирующая тип данных **data_type,** на который ссылается рассматриваемая сущность.

F.3.8.3 Значения

Данный раздел содержит определения неколичественных типов элементов данных (смотри сущности **non_quantitative_int_type** и **non_quantitative_code_type**).

На рисунке F.12 представлены (как модель планирования) основные данные, ассоциированные с неколичественными типами элементов данных.

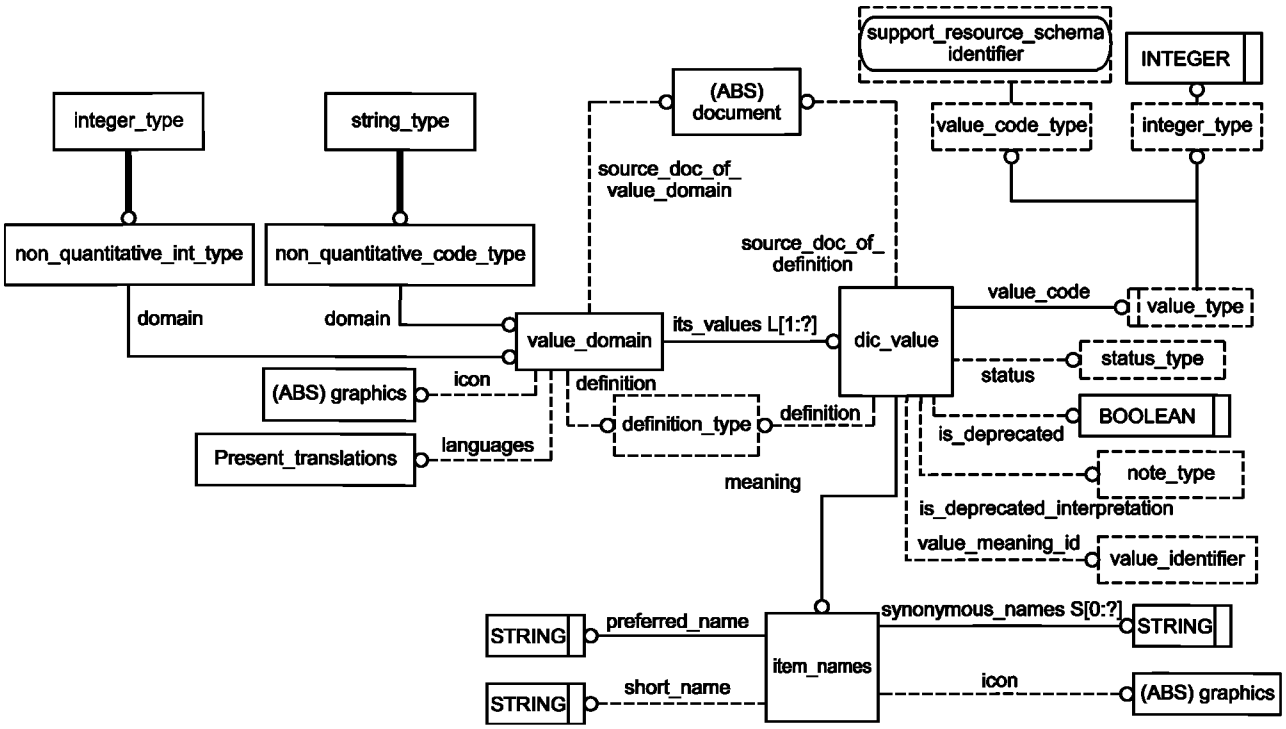


Рисунок 12 — Краткое описание неколичественных типов элементов данных

support_resource_schema identifier	Идентификатор схемы ресурса поддержки
integer	Целое
integer_type	Целый тип
string_type	Строчный тип
(ABS) document	Документ
source_doc_of_value_domain	Исходный документ для области значений
value_code_type	Тип кода значения
integer_type	Целый тип
non_quantitative_int_type	Неколичественный целый тип
non_quantitative_code_type	Неколичественный кодовый тип
source_doc_of_definition	Исходный документ определения
domain	Область
value_domain	Область значений
its_values L[1:?]	Его значения
dic_value	Словарное значение
value_code	Код значения
value_type	Тип значения
(ABS) graphics	Графика
icon	Иконка
definition	Определение
status	Статус
status_type	Тип статуса
present_translations	Настоящие переводы
languages	Языки
definition_type	Тип определения
is_deprecated	Больше не используется
boolean	Булевская величина
meaning	Смысл
is_deprecated_interpretation	Интерпретация величин, не рекомендуемых для использования
node_type	Тип примечания
value_meaning_id	Идентификатор значения величины
value_identifier	Идентификатор значения
string	Строка
preffered_name	Предпочтительное имя
short_name	Короткое имя
item_names	Названия предметов
synonymous_names S[0:?]	Синонимические имена

F.3.8.3.1 Область значений (Value_domain)

Сущность **value_domain** описывает набор допустимых значений неколичественного типа элемента данных.

Пример представления на языке EXPRESS:

```
*)
ENTITY value_domain;
    its_values: LIST [1:?] OF dic_value;
    source_doc_of_value_domain: OPTIONAL document;
    languages: OPTIONAL present_translations;
    terms: LIST [0:?] OF item_names;
    definition: OPTIONAL definition_type;
    icon: OPTIONAL graphics;
WHERE
    WR1: NOT EXISTS(languages) OR (QUERY(v <* its_values |
        languages:<>: v.meaning.languages) = [ ]);
    WR2: codes_are_unique(its_values);
    WR3: EXISTS(languages) OR (QUERY(v <* its_values |
        EXISTS(v.meaning.languages)) = [ ]);
    WR4: EXISTS(languages) OR (QUERY(v <* its_values |
        EXISTS(v.definition.languages)) = [ ]);
END_ENTITY; -- value_domain
(*
```

Определения атрибутов:

its_value: список нумерации значений, содержащихся в описанной области.

source_doc_of_value_domain: документ, описывающий область, ассоциированную с описанной сущностью **value_domain.**

language: вспомогательные языки, на которые выполнены переводы.

terms: список названий предметов **item_names,** дающий для Стандарта IEC 61360 связь со словарем терминов.

definition: вспомогательный текст, описывающий область значений **value_domain.**

icon: вспомогательная иконка **icon,** дающая графическое описание, ассоциированное с областью значений **value_domain.**

Пояснения к тексту программы:

WR1: если смысл значения представлен более, чем на одном языке, то набор используемых языков должен быть одним и тем же для всего набора значений.

WR2: код значения должен быть уникальным внутри указанного типа данных.

WR3: если языки не определены, то смыслу значения язык не назначается.

WR4: если языки не определены, то определению значения язык не назначается.

F.3.8.3.2 Тип значения (Value_type)

Каждое значение неколичественного элемента данных ассоциируется с кодом, характеризующим это значение. Тип значения **value_type** может быть либо целым **integer_type**, либо кодовым **value_code_type.**

Пример представления на языке EXPRESS:

```
*)
TYPE integer_type = INTEGER;
END_TYPE; -- integer_type
TYPE value_type = SELECT(value_code_type, integer_type);
END_TYPE; -- value_type
(*
```

F.3.8.3.3 Словарное значение (Dic_value)

Сущность **dic_value** — это одно из значений сущности **value_domain**.

Пример представления на языке EXPRESS:

```
*)
ENTITY dic_value;
    value_code: value_type;
    meaning: item_names;
    source_doc_of_definition: OPTIONAL document;
    definition: OPTIONAL definition_type;
    status: OPTIONAL status_type;
    is_deprecated: OPTIONAL BOOLEAN;
    is_deprecated_interpretation: OPTIONAL note_type;
    value_meaning_id: OPTIONAL dic_value_identifier;
WHERE
    WR1: NOT EXISTS (SELF. is_deprecated)
        OR EXISTS (SELF. is_deprecated_interpretation);
END_ENTITY; -- dic_value
(*
```

Определения атрибутов:

value_code: код ассоциированный с описанным значением. Это может быть либо кодовый тип **value_code_type,** либо целый тип **integer_type.**

смысл: смысл, ассоциированный с данным значением. Он определяется названием. **source_doc_of_definition:** вспомогательный исходный документ, определяющий рассматриваемое значение.

definition: вспомогательный текст, описывающий словарное значение **dic_value. status:** тип статуса **status_type,** определяющий состояние жизненного цикла словарного значения **dic_value.**

Примечание 1 — Допустимые значения **status_type** не стандартизованы. Они определены для каждого отдельного словаря его поставщиком информации.

Пример 1 — Набор допустимых значений статуса предметов, представленный для стандартизации в агентство технической поддержки стандартов ИСО, определен директивами ИСО.

Пример 2 — Набор допустимых значений статуса предметов в стандартной базе данных МЭК определен директивами МЭК.

Примечание 2 — Если словарные значения **dic_value** еще не подготовлены и не внесены, то может оказаться полезным использование варианта **dic_value**.

Пример 3 — Для эксперимента перед утверждением.

Примечание 3 — Если атрибут status для словарного значения **dic_value** отсутствует, и если данное словарное значение **dic_value** не отменено в соответствии с возможными атрибутом **is_deprecated**, то это рассматриваемое **dic_value** имеет тот же статус стандартизации, что и целый словарь. В частности, если словарь стандартизован, то данное **dic_value** является частью текущего издания стандарта.

is_deprecated: булевская переменная, указывающая (если имеет значение *true*), что рассматриваемое словарное значение **dic_value** использовать уже нельзя.

Примечание 4 — Если атрибут **is_deprecated** значения не имеет, то **dic_value** еще не отменено.

Примечание 5 — Больше не используемые словарные значения **dic_value** сохраняются в области значений **value_domain** по соображениям обеспечения совместимости.

is_deprecated_interpretation: дает обоснование отказа от использования, устанавливает порядок интерпретации значений реализаций больше не используемых элементов.

value_смысл_id: идентификатор словарного значения **dic_value_identifier,** являющийся глобальным идентификатором **dic_value** независимо от области значений **value_domain,** в которую он включен.

Примечание 6 — Данный идентификатор допускает повторное использование рассматриваемого определения словарного значения **dic_value** в различных областях.

Пояснение к тексту программы:

WR1: если атрибут **is_deprecated** существует, то должен существовать и атрибут **is_deprecated_interpretation**.

Дополнительное пояснение к тексту программы:

IP1: значения реализаций элементов **is_deprecated_interpretation** должны быть определены в момент, когда принимается решение об отказе от использования.

F.3.8.3.4 Административные данные (**Administrative_data**)

Сущность **administrative_data** содержит информацию о жизненном цикле словарного элемента.

Пример представления на языке EXPRESS:

```
*)
ENTITY administrative_data;
    status: OPTIONAL status_type;
    translation: LIST[0:?] of translation_data;
    source_language: language_code;
INVERSE
    administrated_element: dictionary_element FOR administration;
WHERE
    WR1: one_language_per_translation(SELF);
    WR2: SIZEOF(QUERY (trans <* SELF.translation |
            trans.language = source_language)) = 0;
END_ENTITY; -- administrative_data
(*
```

Определения атрибутов:

status: тип статуса **status_type,** определяющий состояние жизненного цикла словарного элемента.

Примечание 1 — Допустимые значения типа статуса **status_type** не стандартизованы. Они определены для каждого отдельного словаря его поставщиком информации.

Пример 1 — Набор допустимых значений статуса предметов, предложенный для стандартизации агентством технической поддержки стандарта ИСО, определен директивами ИСО.

Пример 2 — Набор допустимых значений статуса предметов в базе данных стандарта МЭК определен директивами МЭК.

Примечание 2 — Для словарных элементов **dictionary_element**, еще не готовых для практического использования, могут оказаться полезными их предварительные варианты.

Пример 3 — Для экспериментальных целей перед утверждением.

Примечание 3 — Если атрибут **status** отсутствует, и если данный словарный элемент **dictionary_element** больше не используется (смотри атрибут **is_deprecated**), то **dictionary_element** имеет тот же статус стандартизации как и целый словарь. В частности, если словарь стандартизован, то данный **dictionary_element** является частью текущего издания стандарта.

translation: описание надежных трансляторов с различных языков.

source_language: язык, на котором изначально определен словарный элемент **dictionary_element**, на который нужно ссылаться при неувязках трансляции.

Примечание 4 — Словарь может содержать словарные элементы **dictionary_element** с различными исходными языками **source_language**, например, потому что они импортированы из различных словарей. Поставщик словарных данных должен гарантировать, что информация об указанных элементах доступна на рассматриваемом языке или на других языках.

administrated_element: словарный элемент **dictionary_element,** данные жизненного цикла которого имеются в административных данных **administrative_data.**

Пояснения к тексту программы:

WR1: языки перевода, ассоциированные с административными данными **administrative_data,** являются уникальными.

WR2: исходный язык **source_language** не присутствует на языке трансляции, ассоциированном с административными данными **administrative_data.**

F.3.8.3.5 Данные перевода (Translation_data)

Сущность **translation_data** записывает информацию о возможных переводах словарного элемента.

Пример представления на языке EXPRESS:

```
*)
ENTITY translation_data;
    language: language_code;
    responsible_translator: supplier_BSU;
    translation_revision: revision_type;
    date_of_current_translation_revision: OPTIONAL date_type;
INVERSE
    belongs_to: administrative_data FOR translation;
END_ENTITY; -- translation_data
(*
```

Определения атрибутов:

language: язык, на который переведен словарный элемент.

Примечание 1 — В случае невязки между исходным определением словарного элемента и некоторыми его переводами, фактический смысл словарного элемента дается на исходном языке перевода.

responsible_translator: организация, ответственная за трансляцию на язык элемента.

translation_revision: № пересмотра соответствующего перевода.

Примечание 2 — Изменение № версии или изменение № пересмотра словарного элемента не всегда требует каких-либо изменений его перевода. Если изменений перевода, обусловленных изменением № версии или изменением № пересмотра словарного элемента, нет, то соответствующий № пересмотра перевода translation_revision не изменяется. Однако любые изменения перевода подразумевают изменение соответствующего translation_revision.

date_of_current_translation: дата последнего пересмотра перевода.

belongs_to: административные данные **administrative_data,** ссылающиеся на запись данных перевода **translation_data**.

F.3.8.4 Расширения определений единиц измерения по ИСО 10303-41

Данный раздел определяет ресурсы описания единиц измерения в словаре. Здесь даны расширения ресурсов, определенных в ИСО 10303-41.

F.3.8.4.1 Единица, не относящаяся к системе СИ (Non_si_**unit**)

Сущность **non_si_unit** расширяет модель единицы измерения ИСО 10303-41. Она дает представления единиц, не относящихся к системе СИ, которые не являются ни контекстно-зависимыми, ни производными(см. ИСО 10303-41).

Пример представления на языке EXPRESS:

```
*)
ENTITY non_si_unit
SUBTYPE OF(named_unit);
    name: label;
END_ENTITY; -- non_si_unit
(*
```

Определения атрибутов:

names: метки **label,** используемые для именования описанных единиц измерения.

F.3.8.4.2 Правило подтверждения выбора (Assert_ONEOF rule)

Правило **assert_ONEOF** подтверждает, что выбор производится из нижеследующих подтипов поименованных единиц измерения **named_unit:** единицы системы СИ **si_unit,** контекстно-зависимые единицы **context_dependent_unit,** производные единицы **conversion_based_unit** и единицы, не относящиеся к системе СИ **non_si_unit.**

Пример представления на языке EXPRESS:

```
*)
RULE assert_ONEOF FOR(named_unit);
WHERE
    QUERY(u <* named_unit |
    ('ISO13584_IEC61360_DICTIONARY_SCHEMA.NON_SI_Unit'
    IN TYPEOF(u)) AND
    ('MEASURE_SCHEMA.SI_Unit' IN TYPEOF(u))
    OR ('ISO13584_IEC61360_DICTIONARY_SCHEMA.NON_SI_Unit'
    IN TYPEOF(u)) AND
    ('MEASURE_SCHEMA.CONTEXT_DEPENDENT_Unit' IN TYPEOF(u))
    OR ('ISO13584_IEC61360_DICTIONARY_SCHEMA.NON_SI_Unit'
    IN TYPEOF(u)) AND
    ('MEASURE_SCHEMA.CONVERSION_BASED_Unit' IN TYPEOF(u))
    ) = [ ];
END_RULE; — assert_ONEOF
(*
```

F.3.8.4.3 Словарная единица измерения (**Dic_unit**)

Базовое представление единиц измерения производится в структурированной форме в соответствии с ИСО 10303-41. Но так как одной из целей хранения единиц измерения в словаре является удобство пользователя, только одного структурированного представления недостаточно. Оно должно быть дополнено строчным представлением.

Используемые определения имеют варианты:

- можно использовать функцию строки для единицы измерения **string_for_unit** (см. раздел F.3.10). Для рассматриваемого структурированного представления единицы измерения она возвращает строчное представление, соответствующее представлению, используемому в Приложении ВМЭК 61360-1:2009;
- строчное представление может быть дано в обычной текстовой форме (например, сущность математическая строка **mathematical_string,** атрибут текстового представления **text_representation);**
- представление MathML дает расширенное представление единицы измерения, включая верхние индексы, нижние индексы и т.д. (например, сущность математическая строка **mathematical_string,** атрибут представления **MathML_representation).**

Сущность **dic_unit** описывает единицу измерения, включаемую в словарь.

Пример представления на языке EXPRESS:

```
*)
ENTITY dic_unit;
    structured_representation: unit;
    string_representation: OPTIONAL mathematical_string;
END_ENTITY; -- dic_unit
(*
```

Определения атрибутов:

structured_representation: структурированное представление по ИСО 10303-41, включая расширение, определенное в разделе F.3.8.4.

string_representation: функцию **string_for_unit** можно использовать для расчета строчного представления из **structured_representation,** когда строчное представление **string_representation** отсутствует.

Примечание — Атрибут структурированного представления **structured_representation** сущности **dic_unit** используется для кодирования атрибута свойства «Единица измерения».

F.3.9 Определения базового типа и сущности

Данный подраздел содержит определения базового типа и сущности, используемые в основной части модели.

F.3.9.1 Определение базового типа

Нижеследующий раздел содержит определения базового типа и сущности, расположенные по английскому алфавиту.

F.3.9.1.1 Тип кода класса (Class_code_type)

Сущность **class_code_type** задает допустимые значения кода класса.

Пример представления на языке EXPRESS:

```
*)
TYPE class_code_type = code_type;
WHERE
    WR1: LENGTH(SELF) <= class_code_len;
END_TYPE; -- class_code_type
(*
```

WR1: длина значений, соответствующих **class_code_type,** должна быть меньше или равна длине кода класса **class_code_len** (т. е. 35).

F.3.9.1.2 Тип кода (code_type)

Сущность **code_type** задает допустимые значения типа кода, используемого для идентификации.

Примечание — Если код также предназначен для обмена в соответствии с ИСО/ТС 29002-5, то рекомендуется выполнить требования, определенные данным стандартом. Для задания кода можно использовать только «безопасные символы». Безопасные символы включают: буквы верхнего регистра, цифры, двоеточие, десятичную точку, подчеркивание. В некоторых случаях допускается использование символа '–' (минус).

Пример представления на языке EXPRESS:

```
*)
TYPE code_type = identifier;
WHERE
    WR1: NOT(SELF LIKE '*#*');
    WR2: NOT(SELF LIKE '* *');
    WR3: NOT(SELF = '');
END_TYPE; -- code_type
(*
```

Пояснения к тексту программы:

WR1: символ '#' не должен содержаться в значении **code_type**. Символ '#' используется для последовательного соединения идентификаторов (смотри: CONSTANT **sep_id**) или кода и версии (смотри: CONSTANT **sep_cv**).

WR2: пробелы не допускаются (во избежание проблем с предшествующими и последующими пропусками при последовательном соединении кодов).

WR3: значение **code_type** не должно быть пустой строкой.

F.3.9.1.3 Тип валюты (Currency_code)

Сущность **currency_code** задает значения, допустимые для кода валюты. Указанные значения соответствуют ИСО 4217.

Пример — «CHF» — код швейцарского франка, «CNY» — код китайского юаня, «JPY» — код японской йены, «SUR» — код рубля СССР, «USD» — код доллара США, «EUR» — код евро.

Пример представления на языке EXPRESS:

```
*)
TYPE currency_code = identifier;
WHERE
    WR1: LENGTH(SELF) = 3;
END_TYPE; -- currency_code
(*
```

Пояснения к тексту программы:

WR1: длина кода валюты **currency_code** равна 3.

F.3.9.1.4 Тип кода типа данных (**Data_type**_code_type)

Сущность **data_type_code_type** задает значения, допустимые для кода типа данных.

Пример представления на языке EXPRESS:

```
*)
TYPE data_type_code_type = code_type;
WHERE
    WR1: LENGTH(SELF) = data_type_code_len;
END_TYPE; -- data_type_code_type
(*
```

Пояснения к тексту программы:

WR1: длина типа кода типа данных **data_type_code_type** должна быть равна значению **data_type_code_len** (т. е. 35).

F.3.9.1.5 Тип даты (Date_type)

Сущность **date_type** задает значения, допустимые для даты. Указанные значения соответствуют ИСО 8601.

Пример — «1994-03-21».

Пример представления на языке EXPRESS:

```
        *)
        TYPE date_type = STRING(10) FIXED;
        WHERE
            WR1: SELF LIKE '#### - ## - ##';
        END_TYPE; -- date_type
    (*
```

F.3.9.1.6 Тип определения (Definition_type)

Сущность **definition_type** задает значения, допустимые для определения типа.

Пример представления на языке EXPRESS:

```
        *)
        TYPE definition_type = translatable_text;
        END_TYPE; -- definition_type
        (*
```

F.3.9.1.7 Тип классификации типа элемента данных (DET_classification_type)

Сущность **DET_classification_type** задает значения, допустимые для классификации типов элементов данных **DET**. Указанные значения используются для классификации **DET** в соответствии с ИСО 80000/МЭК 80000 (ранее ИСО 31).

Пример представления на языке EXPRESS:

```
        *)
        TYPE DET_classification_type = identifier;
        WHERE
            WR1: LENGTH(SELF) = DET_classification_len;
        END_TYPE; -- DET_classification_type
        (*
```

Пояснения к тексту программы:

WR1: длина значения **DET_classification_type** должна быть равна значению **DET_classification_len** (т. е. 3**).**

F.3.9.1.8 Тип примечания (Note_type)

Сущность **note_type** задает значения, допустимые для примечаний.

Пример представления на языке EXPRESS:

```
        *)
        TYPE note_type = translatable_text;
        END_TYPE; -- note_type
        (*
```

F.3.9.1.9 Тип предпочтительного имени (Pref_name_type)

Сущность **pref_name_type** задает значения, допустимые для предпочтительного имени.

Пример представления на языке EXPRESS:

```
        *)
        TYPE pref_name_type = translatable_label;
        WHERE
            WR1: check_label_length(SELF, pref_name_len);
        END_TYPE; -- pref_name_type
        (*
```

Пояснения к тексту программы:

WR1: длина значения **pref_name_type** не должна превышать длину **pref_name_len** (т. е. 255).

F.3.9.1.10 Тип кода свойства (Property_code_type)

Сущность **property_code_type** задает значения, допустимые для кода свойства.

Пример представления на языке EXPRESS:

```
        *)
        TYPE property_code_type = code_type;
        WHERE
            WR1: LENGTH(SELF) <= property_code_len;
        END_TYPE; -- property_code_type
        (*
```

Пояснения к тексту программы:

WR1: длина значения **property_code_type** не должна превышать длину **property_code_len** (т. е. 35).

F.3.9.1.11 Тип заметки (Remark_type)

Сущность **remark_type** задает значения, допустимые для заметки.

Пример представления на языке EXPRESS:

```
        *)
        TYPE remark_type = translatable_text;
        END_TYPE; -- remark_type
        (*
```

F.3.9.1.12 Тип иерархического положения (Hierarchical_position_type)

Сущность **hierarchical_position_type** задает значения, допустимые для иерархического положения.

Пример представления на языке EXPRESS:

```
*)
TYPE hierarchical_position_type = identifier;
END_TYPE; -- hierarchical_position_type
(*
```

Примечание — Представление иерархического положения атрибутом **hierarchical_position_type** основано на использовании некоторых конвенций кодирования. В настоящем стандарте конвенции кодирования не рассматриваются.

F.3.9.1.13 Тип пересмотра (Revision_type)

Сущность **revision_type** задает значения, допустимые для № пересмотра.

Примечание — Для новой версии № пересмотра равен '0'.

Пример представления на языке EXPRESS:

```
*)
TYPE revision_type = code_type;
WHERE
    WR1: LENGTH(SELF) <= revision_len;
    WR2: EXISTS(VALUE(SELF)) AND ('INTEGER' IN TYPEOF(VALUE(SELF)))
        AND (VALUE(SELF) >= 0);
END_TYPE; -- revision_type
(*
```

Пояснения к тексту программы:

WR1: длина значения **revision_type** не должна превышать длину **revision_len** (т. е. 3).

WR2: значение **revision_type** должно содержать только цифры, его целое значение может быть только натуральным.

F.3.9.1.14 Тип короткого имени (Short_name_type)

Сущность **short_name_type** задает значения, допустимые для короткого имени.

Пример представления на языке EXPRESS:

```
*)
TYPE short_name_type = translatable_label;
WHERE
    WR1: check_label_length(SELF, short_name_len);
END_TYPE; -- short_name_type
(*
```

Пояснения к тексту программы:

WR1: длина значения **short_name_type** не должна превышать длину **short_name_len** (т. е. 30).

F.3.9.1.15 Тип кода поставщика (Supplier_code_type)

Сущность **supplier_code_type** задает значения, допустимые для кода поставщика.

Примечание — Если код поставщика также предназначен для обмена в соответствии с ИСО/ТС 29002-5, то различные части кода поставщика в соответствии с ИСО 6523(ICD, OI, OPI, OPIS и AI) разделяются символом '-'.

Пример представления на языке EXPRESS:

```
*)
TYPE supplier_code_type = code_type;
WHERE
    WR1: LENGTH(SELF) <= supplier_code_len;
END_TYPE; -- supplier_code_type
(*
```

Пояснения к тексту программы:

WR1: длина значения **supplier_code_type** не должна превышать длину **supplier_code_len** (т. е. 149).

F.3.9.1.16 Тип синонимического имени (Syn_name_type)

Сущность **syn_name_type** задает значения, допустимые для синонимического имени.

Пример представления на языке EXPRESS:

```
*)
TYPE syn_name_type = SELECT(label_with_language, label);
WHERE
    WR1: check_syn_length(SELF, syn_name_len);
END_TYPE; -- syn_name_type
(*
```

Пояснения к тексту программы:

WR1: длина значения **syn_name_type** не должна превышать длину **syn_name_len** (т. е. 255).

F.3.9.1.17 Тип ключевого слова (Keyword_type)

Сущность **keyword_type** задает значения, допустимые для ключевых слов.

Пример представления на языке EXPRESS:

```
*)
TYPE keyword_type = SELECT(label_with_language, label);
END_TYPE; -- keyword_type
(*
```

F.3.9.1.18 Тип глобальной идентификации ISO_29002_IRDI_type

Сущность **ISO_29002_IRDI_type** — это глобальный идентификатор, указывающий на администрируемый предмет в реестре. Структура данного идентификатора удовлетворяет синтаксическим требованиям к идентификаторам, определенным в ИСО/ТС 29002-5.

Примечание 1 — Сущность **ISO_29002_IRDI_type** может быть использована для любого вида информации, рассмотренной в ИСО/ТС 29002-5 и ассоциированной с идентификатором МИРД. Ниже рассмотрены три специальных случая, нашедшие применение в стандартной словарной схеме **ISO13584_IEC61360_dictionary_schema**: идентификатор ограничения **constraint_identifier**, идентификатор словарной единицы измерения **dic_unit_id**entifier и идентификатор словарного значения **dic_value_identifier**.

Пример представления на языке EXPRESS:

```
*)
TYPE ISO_29002_IRDI_type = identifier;
WHERE
        WR1: LENGTH (SELF) <= 290;
END_TYPE; -- syn_name_type
(*
```

Пояснения к тексту программы:

WR1: в соответствии с ИСО/ТС 29002-5 длина идентификатора не должна превышать 290.

Дополнительные пояснения:

IP1: идентификатор должен удовлетворять требованиям, установленным в ИСО/ТС 29002-5 для «международного идентификатора регистрации данных» (МИРД).

Примечание 2 — В соответствии с ИСО/ТС 29002-5 МИРД состоит либо из строки, не содержащей символ '#' (идентификатор организации), либо из трех подстрок, не содержащих символ '#' внутри и разделенных символом '#' для указания администрированных предметов.

Примечание 3 — Случаи, когда МИРД не используется для идентификации организации:

- первая подстрока, называемая Регистрационным Идентификатором Полномочий (RAI), указывает организацию, несущую ответственность за администрирование рассматриваемого предмета;
- вторая подстрока, называемая идентификатором данных (DI), содержит как категоризацию администрируемого предмета, представленную двумя символами и последующим знаком минус ('-') в соответствии с ИСО/ТС 29002-5 (например, класс, свойство, единица измерения), так и идентификатор, назначенный для администрируемого предмета регистрационным идентификатором полномочий RAI;
- третья подстрока соответствует Идентификатору версии (VI) МИРД.

F.3.9.1.19 Идентификатор ограничения (Constraint_identifier)

Идентификатор **constraint_identifier** — это идентификатор типа **ISO_29002_IRDI_type**, дающий глобальное указание на ограничение, представленное как сущность **constraint.** Структура данного идентификатора удовлетворяет синтаксическим требованиям, определенным в ИСО/ТС 29002-5.

Примечание — Идентификатор ограничения **constraint_identifier** может быть ассоциирован с услугой разрешения по ИСО/ТС 29002-20. Данная услуга дает возможность получить формальное определение ограничения, идентифицированного как сущность **constraint_identifier** в соответствии с моделью ограничения EXPRESS в рамках синтаксиса, определенного ИСО 13584-32 (OntoML) и, возможно, ИСО 10303-21.

Пример представления на языке EXPRESS:

```
*)
TYPE constraint_identifier = ISO_29002_IRDI_type;
END_TYPE; -- constraint_identifier
(*
```

Дополнительные пояснения:

IP1: часть идентификатора, идущая после второго символа '#' (идентификатор данных), должна начинаться с '04-', чтобы указать на ограничение в соответствии с ИСО/ТС 29002-5.

F.3.9.1.20 Идентификатор словарной единицы измерения (**Dic_unit_id**entifier)

Идентификатор **dic_unit_identifier** — это идентификатор **ISO_29002_IRDI_type,** задающий единицу измерения, представление которой в формате **dic_unit** можно загрузить с сервера ИСО/ТС 29002-20. Структура данного идентификатора удовлетворяет синтаксическим требованиям ИСО/ТС 29002-5.

Пример представления на языке EXPRESS:

```
*)
TYPE dic_unit_identifier = ISO_29002_IRDI_type;
END_TYPE; -- dic_unit_identifier
(*
```

Дополнительные пояснения:

IP1: идентификатор **dic_unit_identifier** должен быть ассоциирован с услугой разрешения по ИСО/ТС 29002. Данная услуга предоставляет формальное определение единицы измерения, идентифицированной сущностью **dic_unit_identifier,** соответствующей требованиям модели **dic_unit** языка EXPRESS в синтаксисе, определенном ИСО 13584-32 (OntoML) и, возможно, ИСО 10303-21.

IP2: часть идентификатора, идущая после второго символа '#' (идентификатор данных), должна начинаться с '05-', чтобы указать на единицу измерения по ИСО/ТС 29002-5.

Примечание — Идентификатор словарной единицы измерения **dic_unit_identifier** составляет Международный идентификатор регистрации данных (МИРД) в соответствии с ИСО/МЭК 11179-5.

F.3.9.1.21 Идентификатор словарного значения (Dic_value_identifier)

Идентификатор **dic_value_identifier** — это стандартный идентификатор типа **ISO_29002_IRDI_type**, дающий глобальный идентификатор значения свойства, представленный как сущность **dic_value**. Структура данного идентификатора удовлетворяет синтаксическим требованиям ИСО/ТС 29002-5.

Примечание 1 — Назначение идентификатора словарного значения **dic_value_identifier** допускает повторное использование рассматриваемого определения **dic_value** в нескольких областях значений **value_domain**.

Примечание 2 — Идентификатор **dic_value_identifier** может быть ассоциирован с услугой разрешения по ИСО/ТС 29002. Данная услуга дает возможность получить формальное определение значения, идентифицированного сущностью **dic_value_identifier** в соответствии с моделью **dic_value** языка EXPRESS в синтаксисе, определенном ИСО 13584-32 (OntoML) и, возможно, ИСО 10303-21.

Пример представления на языке EXPRESS:

```
*)
TYPE dic_value_identifier = ISO_29002_IRDI_type;
END_TYPE; -- dic_value_identifier
(*
```

Дополнительное пояснение к тексту программы:

IP1: часть идентификатора, идущая после второго символа '#' (идентификатор данных), должна начинаться с '07-' для указания значения свойства в соответствии с ИСО/ТС 29002-5.

Примечание 3 — Идентификатор словарного значения **dic_value_identifier** составляет Международный идентификатор регистрации данных (МИРД) в соответствии с ИСО/МЭК 11179-5.

F.3.9.1.22 Тип кода значения (Value_code_type)

Сущность **value_code_type** задает значения, допустимые для кода значения.

Пример представления на языке EXPRESS:

```
*)
TYPE value_code_type = identifier;
WHERE
    WR1: LENGTH(SELF) <= value_code_len;
END_TYPE; -- value_code_type
(*
```

Пояснения к тексту программы:

WR1: длина значения **value_code_type** не должна превышать длину **value_code_len** (т. е. 35).

F.3.9.1.23 Тип формата значения (Value_format_type)

Сущность **value_format_type** задает значения, допустимые для формата значения. Указанные значения определяются в соответствии с приложением D.

Пример представления на языке EXPRESS:

```
*)
TYPE value_format_type = identifier;
WHERE
    WR1: LENGTH(SELF) <= value_format_len;
END_TYPE; -- value_format_type
(*
```

Пояснения к тексту программы:

WR1: длина значения **value_format_type** не должна превышать длину **value_format_len** (т. е. 80).

F.3.9.1.24 Тип версии (Version_type)

Сущность **version_type** задает значения, допустимые для № версии.

Пример представления на языке EXPRESS:

```
*)
TYPE version_type = code_type;
WHERE
    WR1: LENGTH(SELF) <= version_len;
    WR2: EXISTS(VALUE(SELF)) AND ('INTEGER' IN TYPEOF(VALUE(SELF)))
```

```
            AND (VALUE(SELF) >= 0);
    END_TYPE; -- version_type
    (*
```

Пояснения к тексту программы:

WR1: длина значения **version_type** не должна превышать длину **version_len** (т. е. 10).

WR2: значение **version_type** должно содержать только цифры.

F.3.9.1.25 Тип статуса (Status_type)

Сущность **status_type** задает значения, допустимые для статуса. Допустимые значения **status_type** не стандартизованы. Они должны быть определены для каждого отдельного словаря поставщиком словарных данных.

Пример 1 — Набор допустимых значений статуса предметов, предлагаемый для стандартизации агентству технической поддержки стандарта ИСО, определен директивами ИСО.

Пример 2 — Набор допустимых значений статуса предметов в базе данных стандарте МЭК определен директивами МЭК.

Примечание — Статус может быть ассоциирован со словарным элементом **dictionary_element** или со словарным значением **dic_value.**

Пример представления на языке EXPRESS:

```
    *)
    TYPE status_type = identifier;
    END_TYPE; -- status_type
    (*
```

F.3.9.1.26 Тип кода словаря (Dictionary_code_type)

Сущность **dictionary_code_type** — это тип кода **code_type,** задающий словарь.

Пример представления на языке EXPRESS:

```
    *)
    TYPE dictionary_code_type = identifier;
    WHERE
        WR1: LENGTH(SELF) <= dictionary_code_len;
    END_TYPE; -- dictionary_code_type
    (*
```

Пояснения к тексту программы:

WR1: длина значения **dictionary_code_type** не должна превышать длину **dictionary_code_len** (т. е. 131).

F.3.9.2 Определения базовой сущности

Данный подраздел содержит определения базовой сущности, расположенные по английскому алфавиту.

F.3.9.2.1 Даты (dates)

Сущность **dates** описывает три даты, ассоциированные соответственно с первым устойчивым описанием, текущим № версии и текущим № пересмотра для данного описания.

Примечание — По каждому отдельному правилу управления словарем поставщик информации словаря несет ответственность за выбор момента времени первого устойчивого описания предмета.

Пример представления на языке EXPRESS:

```
    *)
    ENTITY dates;
        date_of_original_definition: date_type;
        date_of_current_version: date_type;
        date_of_current_revision: OPTIONAL date_type;
    END_ENTITY; -- dates
    (*
```

Определения атрибутов:

date_of_initial_definition: дата, ассоциированная с первой устойчивой версией предмета.

date_of_current_version: дата, ассоциированная с текущей версией.

date_of_current_revision: дата, ассоциированная с последним пересмотром.

F.3.9.2.2 Документ (document)

Сущность **document** — это абстрактный ресурс, заменяющий документ. Словарная схема обеспечивает только обмен идентификации документа (смотри ниже). Сущность **document** может также быть описана как подтип с сущностями, использующими средства обмена данных документа.

Пример — Путем ссылки на внешний файл и на точную спецификацию формата файла.

Пример представления на языке EXPRESS:

```
    *)
    ENTITY document
    ABSTRACT SUPERTYPE;
    END_ENTITY; -- document
    (*
```

F.3.9.2.3 Графика (graphics)

Сущность **graphics** — это абстрактный ресурс, описываемый как подтип, содержащий сущности, применяющие средства обмена графическими данными.

Пример — Ссылка на внешний файл и на точное задание формата файла.

Пример представления на языке EXPRESS:

```
*)
ENTITY graphics
ABSTRACT SUPERTYPE;
END_ENTITY; -- graphics
(*
```

F.3.9.2.4 Внешняя графика (External_graphics)

Сущность **external_graphics** обеспечивает обмен графических данных с помощью внешних файлов, на которые производится ссылка сущностью **graphic_file**.

Пример представления на языке EXPRESS:

```
*)
ENTITY external_graphics
SUBTYPE OF (graphics);
        representation: graphic_files;
END_ENTITY; -- external_graphics
(*
```

Определения атрибутов:

representation: представление графики с помощью внешних файлов.

F.3.9.2.5 Графический файл (Graphic_file)

Графический файл **graphic_file** — это внешний предмет **external_item,** содержанием которого является рисунок.

Примечание 1 — Сущность **external_item**, определенная в ИСО 13584-24:2003, — это предмет, содержание которого может быть составлено из библиотечных внешних файлов. Она ссылается на протокол внешнего файла **external_file_protocol**, указывающий порядок обработки библиотечного внешнего файла, и на внешний контент **external_content**, указывающий библиотечные внешние файлы, представляющие нужное содержание.

Примечание 2 — И протокол **external_file_protocol**, и контент **external_content** определены в ИСО 13584-24:2003.

Примечание 3 — Только на внешний контент **external_content**, содержащий **http_file**, и только на гипертекстовые протоколы **http_protocol** и протоколы внешних файлов **external_file_protocols** производится ссылка стандартной словарной схемой **ISO13584_IEC61360_dictionary_schema**. Указанные сущности могут быть использованы в контексте настоящего стандарта.

Примечание 4 — Графические файлы **graphic_file** могут зависеть от языка. Это определяется нижеследующими подтипами внешнего контента **external_content**: непереводимый внешний контент **not_translatable_external_content**, непереведенный внешний контент **not_translated_external_content**, переведенный внешний контент **translated_external_content**.

Примечание 5 — Сущности **http_file**, **http_protocol**, **not_translatable_external_content**, **not_translated_external_content** и и определены в ИСО 13584-24:2003. На них производится ссылка стандартной словарной схемой **ISO13584_IEC61360_dictionary_schema**.

Пример представления на языке EXPRESS:

```
*)
ENTITY graphic_files
SUBTYPE OF (external_item);
END_ENTITY; -- graphic_files
(*
```

F.3.9.2.6 Идентифицированный документ (Identified_document)

Сущность **identified_document** описывает документ, идентифицированный своей меткой.

Пример представления на языке EXPRESS:

```
*)
ENTITY identified_document
SUBTYPE OF(document);
        document_identifier: translatable_label;
WHERE
        WR1: check_label_length(SELF.document_identifier,source_doc_len);
END_ENTITY; -- identified_document
(*
```

Определения атрибутов:

document_identifier: метка описанного документа.

Пояснения к тексту программы:

WR1: длина значения **document_identifier** не должна превышать длину **source_doc_len** (т. е. 255).

F.3.9.2.7 Названия предметов (Item_names)

Сущность **item_names** задает имена, которые можно ассоциировать с данным описанием. Она определяет предпочтительное имя, набор синонимических имен, короткое имя и языки для прочих имен. Данная сущность может быть ассоциирована с иконкой.

Пример представления на языке EXPRESS:

```
*)
ENTITY item_names;
        preferred_name: pref_name_type;
        synonymous_names: SET OF syn_name_type;
        short_name: OPTIONAL short_name_type;
        languages: OPTIONAL present_translations;
        icon: OPTIONAL graphics;
WHERE
        WR1: NOT(EXISTS(languages )) OR (
                ('ISO13584_IEC61360_LANGUAGE_RESOURCE_SCHEMA'
                + '.TRANSLATED_LABEL' IN TYPEOF(preferred_name))
                AND (languages:=:
                preferred_name\translated_label.languages)
                AND (NOT(EXISTS(short_name)) OR
                ('ISO13584_IEC61360_LANGUAGE_RESOURCE_SCHEMA'
                + '.TRANSLATED_LABEL' IN TYPEOF(short_name))
                AND (languages:=: short_name\translated_label.languages))
                AND (QUERY(s <* synonymous_names |
                        NOT('ISO13584_IEC61360_DICTIONARY_SCHEMA' +
                '.LABEL_WITH_LANGUAGE' IN TYPEOF(s))) = [ ]));
                WR2: NOT EXISTS(languages) OR (QUERY(s <* synonymous_names |
                EXISTS(s.language) AND NOT(s.language IN
                QUERY(l <* languages.language_codes | TRUE
                ))) = [ ]);
        WR3: EXISTS(languages) OR
                (('SUPPORT_RESOURCE_SCHEMA.LABEL' IN
                TYPEOF(preferred_name))
                AND (NOT(EXISTS(short_name)) OR
                ('SUPPORT_RESOURCE_SCHEMA.LABEL' IN
                TYPEOF(short_name)))
                AND (QUERY(s <* synonymous_names |
                'ISO13584_IEC61360_DICTIONARY_SCHEMA.LABEL_WITH_LANGUAGE' IN
                TYPEOF(s)) = [ ] ));
END_ENTITY; -- item_names
(*
```

Определения атрибутов:

preferred_name: имя, предпочтительное для использования.

synonymous_names: набор синонимических имен.

short_names: аббревиатура предпочтительного имени.

languages: список языков по выбору, на которых даются другие имена.

icon: вспомогательная иконка для графического представления описания, ассоциированного с атрибутом **item_names**.

Пояснения к тексту программы:

WR1: если предпочтительные и короткие имена даются более чем на одном языке, то все атрибуты **languages** переведенных меток **translated_labels** должны содержать реализации переводов **present_translations** как в атрибуте **languages** рассматриваемой реализации **item_names**.

WR2: если синонимические имена даются на более чем одном языке, то можно использовать только языки, указанные в реализациях переводов **present_translations** и в атрибутах **languages** рассматриваемой реализации **item_names**.

WR3: если язык не указан, то атрибуты **preferred_names, short_names** и **synonymous_names** можно не переводить.

Примечание 1 — Атрибуты **preferred_names**, **synonymous_names** и **short_names** используются для кодирования атрибутов «Предпочтительное имя», «Синонимическое имя» и «Короткое имя» свойств и классов соответственно.

Примечание 2 — Атрибут **languages** используется для определения последовательности переводов (если это необходимо для атрибутов).

F.3.9.2.8 Метка с языком (Label_with_language)

Сущность **label_with_language** обеспечивает ресурс, ассоциирующий метку с языком.

Пример представления на языке EXPRESS:

```
*)
ENTITY label_with_language;
    l: label;
    language: language_code;
END_ENTITY; -- label_with_language
(*
```

Определения атрибутов:

l: метка, ассоциированная с языком.

language: код помеченного языка.

F.3.9.2.9 Математическая строка (Mathematical_string)

Сущность **mathematical_string** обеспечивает ресурсы, определяющие представление математических строк. Она допускает представление в формате MathML.

Пример представления на языке EXPRESS:

```
*)
ENTITY mathematical_string;
    text_representation: text;
    MathML_representation: OPTIONAL text;
END_ENTITY; -- mathematical_string
(*
```

Определения атрибутов:

text_representation: «линейная» форма математической строки, использующая ИСО 843 (при необходимости).

MathML_representation: текст в формате MathML, размеченный в соответствии с требованиями XML DTD (определения типа документа) для MathML. Текст MathML обрабатывается как отдельная строка во время обмена (см. ИСО 10303-21).

F.3.10 Определения функций

Данный подраздел содержит функции, на которые производится ссылка из разделов по месту для подтверждения непротиворечивости данных, и которые доставляют ресурсы для разработки приложений.

F.3.10.1 Функция ациклического соотношения суперклассов (Acyclic_superclass_relationship)

Функция **acyclic_superclass_relationship** проверяет отсутствие цикла в соотношении суперклассов. Кардинальное число атрибута **its_superclass** в классе сущностей гарантирует, что существует дерево наследственности, а ациклических графов нет. То есть, данная функция просто не проверяет тот факт, что реализации класса не ссылаются (в атрибуте **its_superclass**) на другую реализацию, которая, в сущности, является подклассом.

Пример представления на языке EXPRESS:

```
*)
FUNCTION acyclic_superclass_relationship(
    current: class_BSU; visited: SET OF class): LOGICAL;
IF SIZEOF(current.definition) = 1 THEN
    IF current.definition[1] IN visited THEN
        RETURN(FALSE);
    (* wrong: current declares a subclass as its superclass    *)
    ELSE
        IF EXISTS(current.definition[1]\class.its_superclass)
        THEN
            RETURN(acyclic_superclass_relationship(
                current.definition[1]\class.its_superclass,
                visited + current.definition[1]));
        ELSE
            RETURN(TRUE);
        END_IF;
    END_IF;
ELSE
    RETURN(UNKNOWN);
END_IF;
END_FUNCTION; -- acyclic_superclass_relationship
(*
```

F.3.10.2 Функция проверки длины (Check_syn_length)

Функция **check_syn_length** проверяет тот факт, что длина атрибута **s** не превышает длину, определенную атрибутом **s_length.**

Пример представления на языке EXPRESS:

```
*)
FUNCTION check_syn_length(s: syn_name_type; s_length: INTEGER):BOOLEAN;
IF 'ISO13584_IEC61360_DICTIONARY_SCHEMA.LABEL_WITH_LANGUAGE'
    IN TYPEOF(s)
THEN
    RETURN(LENGTH(s.l) <= s_length);
ELSE
    RETURN(LENGTH(s) <= s_length);
END_IF;
END_FUNCTION; -- check_syn_length
(*
```

F.3.10.3 Функция проверки уникальности кода (Codes_are_unique)

Функция **codes_are_unique** возвращает значение TRUE, если коды значений **value_codes** уникальны внутри рассматриваемого списка значений.

Пример представления на языке EXPRESS:

```
*)
FUNCTION codes_are_unique(values: LIST OF dic_value): BOOLEAN;
LOCAL
    ls: SET OF STRING:= [ ];
    li: SET OF INTEGER:= [ ];
END_LOCAL;

IF('ISO13584_IEC61360_DICTIONARY_SCHEMA.VALUE_CODE_TYPE' IN
    TYPEOF(values[1].value_code))
THEN
        REPEAT i:= 1 TO SIZEOF(values);
            ls:= ls + values[i].value_code;
        END_REPEAT;

        RETURN(SIZEOF(values) = SIZEOF(ls));
    ELSE
        IF('ISO13584_IEC61360_DICTIONARY_SCHEMA.INTEGER_TYPE' IN
            TYPEOF(values[1].value_code))
        THEN
            REPEAT i:= 1 TO SIZEOF(values);
                li:= li + values[i].value_code;
            END_REPEAT;
        RETURN(SIZEOF(values) = SIZEOF(li));
    ELSE
        RETURN(?);
    END_IF;
END_IF;
END_FUNCTION; -- codes_are_unique
(*
```

F.3.10.4 Функция проверки наличия определения (Definition_available_implies)

Функция **definition_available_implies** проверяет, действительно ли существует определение, соответствующее рассматриваемому параметру **БСЕ**. Если данное определение существует, то возвращается значение параметра **expression**.

Пример представления на языке EXPRESS:

```
*)
FUNCTION definition_available_implies(
    BSU: basic_semantic_unit;
    expression: LOGICAL): LOGICAL;
RETURN(NOT(SIZEOF(BSU.definition) = 1) OR expression);

END_FUNCTION; -- definition_available_implies
(*
```

F.3.10.5 Функция проверки подкласса (Is_subclass)

Функция **is_subclass** возвращает значение TRUE, если подкласс является либо суперклассом, либо подклассом суперкласса. Если некоторые словарные определения **dictionary_definition** класса недоступны, то функция возвращает значение UNKNOWN.

Пример представления на языке EXPRESS:

```
*)
FUNCTION is_subclass(sub, super: class): LOGICAL;
    IF (NOT EXISTS(sub)) OR (NOT EXISTS(super)) THEN
        RETURN(UNKNOWN);
    END_IF;
    IF sub = super
    THEN
        RETURN(TRUE);
    END_IF;

    IF NOT EXISTS(sub.its_superclass)
    THEN
        (* end of chain reached, didn't meet super so far    *)
        RETURN(FALSE);
    END_IF;
    IF SIZEOF(sub.its_superclass.definition) = 1
    THEN
    (* definition available      *)
        IF (sub.its_superclass.definition[1] = super)
        THEN
            RETURN(TRUE);
        ELSE
            RETURN(is_subclass(sub.its_superclass.definition[1], super));
        END_IF;
    ELSE
        RETURN(UNKNOWN);
    END_IF;
END_FUNCTION; -- is_subclass
(*
```

F.3.10.6 Функция строчного представления производной единицы измерения (String_for_derived_**unit**)

Функция **string_for_derived_unit** возвращает строчное представление производной единицы измерения **derived_unit** (в соответствии с ИСО 10303-41), рассматриваемой как параметр. Элементы производной единицы измерения отличаются по знаку показателя степени. Если существуют элементы обоих знаков, то обозначение *'/'* используется для отделения положительных значений от отрицательных. Если имеются только отрицательные показатели степени, то используется обозначение «u-e». Точка '.' (десятичный код №46 в соответствии с ИСО/МЭК 8859-1, см. раздел ИСО 10303-21) используется для отделения индивидуальных элементов.

Пример представления на языке EXPRESS:

```
*)
FUNCTION string_for_derived_unit(u: derived_unit): STRING;

    FUNCTION string_for_derived_unit_element(
        u: derived_unit_element; neg_exp: BOOLEAN
        (* print negative exponents with power –1     *)): STRING;
        (* returns a STRING representation of the
            derived_unit_element (according to ISO 10303-41)
            passed as parameter      *)
LOCAL
    result: STRING;
END_LOCAL;

result:= string_for_named_unit(u.unit);
IF (u.exponent <> 0)
THEN
    IF (u.exponent > 0) OR NOT neg_exp
    THEN
        result:= result + '**' + FORMAT(
            ABS(u.exponent), '2I')[2];
    ELSE
        result:= result + '**' + FORMAT(u.exponent, '2I')[2];
    END_IF;
END_IF;
    RETURN(result);
END_FUNCTION; -- string_for_derived_unit_element
```

```
LOCAL
        pos, neg: SET OF derived_unit_element;
        us: STRING;
END_LOCAL;

(* separate unit elements according to the sign of the exponents:        *)
pos:= QUERY(ue <* u.elements | ue.exponent > 0);
neg:= QUERY(ue <* u.elements | ue.exponent < 0);
us:= '';
IF SIZEOF(pos) > 0 THEN
        (* there are unit elements with positive sign   *)
        REPEAT i:= LOINDEX(pos) TO HIINDEX(pos);
                us:= us + string_for_derived_unit_element(pos[i], FALSE);
                IF i <> HIINDEX(pos)
                THEN
                        us:= us + '.';
                END_IF;
        END_REPEAT;

        IF SIZEOF(neg) > 0
        THEN

                        (* there are unit elements with negative sign, use '/'notation:      *)
                        us:= us + '/';
                        IF SIZEOF(neg) > 1
                        THEN
                                us:= us + '(';
                        END_IF;

                        REPEAT i:= LOINDEX(neg) TO HIINDEX(neg);
                                us:= us + string_for_derived_unit_element(
                                        neg[ i], FALSE);
                                IF i <> HIINDEX(neg)
                                THEN
                                        us:= us + '.';
                                END_IF;
                        END_REPEAT;

                        IF SIZEOF(neg) > 1
                        THEN
                                us:= us + ')';
                        END_IF;
                END_IF;
ELSE
                (* only negative signs, use u-e notation        *)
                IF SIZEOF(neg) > 0 THEN
                        REPEAT i:= LOINDEX(neg) TO HIINDEX(neg);
                                us:= us + string_for_derived_unit_element(
                                        neg[ i], TRUE);
                                IF i <> HIINDEX(neg)
                                THEN
                                        us:= us + '.';
                                END_IF;
                        END_REPEAT;
                END_IF;
        END_IF;
        RETURN(us);

        END_FUNCTION; -- string_for_derived_unit
        (*
```

F.3.10.7 Функция строчного представления поименованных единиц измерения (String_for_named_**unit**)

Функция **string_for_named_unit** возвращает строчное представление поименованной единицы измерения **named_unit** (в соответствии с ИСО 10303-41 и расширении в 5.10.6.2), рассматриваемой как параметр.

Пример представления на языке EXPRESS:

```
*)
FUNCTION string_for_named_unit(u: named_unit): STRING;

IF 'MEASURE_SCHEMA.SI_Unit' IN TYPEOF(u) THEN
    RETURN(string_for_SI_unit(u));
ELSE
    IF 'MEASURE_SCHEMA.CONTEXT_DEPENDENT_Unit' IN TYPEOF(u)
        THEN
            RETURN(u\context_dependent_unit.name);
        ELSE
            IF 'MEASURE_SCHEMA.CONVERSION_BASED_Unit' IN TYPEOF(u)
            THEN
                RETURN(u\conversion_based_unit.name);
            ELSE
                IF 'ISO13584_IEC61360_DICTIONARY_SCHEMA'
                    +'.NON_SI_Unit' IN TYPEOF(u)
            THEN
                RETURN(u\non_si_unit.name);
            ELSE
                RETURN('name_unknown');
            END_IF;
        END_IF;
    END_IF;
END_IF;
END_FUNCTION; -- string_for_named_unit
(*
```

F.3.10.8 Функция строчного представления единицы системы СИ (String_for_SI_**unit**)

Функция **string_for_SI_unit** возвращает строчное представление единицы системы СИ **si_unit** (в соответствии с ИСО 10303-41), рассматриваемой как параметр.

Пример представления на языке EXPRESS:

```
*)
FUNCTION string_for_SI_unit(unit: si_unit): STRING;

LOCAL
    prefix_string, unit_string: STRING;
END_LOCAL;

IF EXISTS(unit.prefix) THEN
    CASE unit.prefix OF
        exa: prefix_string:= 'E';
        peta: prefix_string:= 'P';
        tera: prefix_string:= 'T';
        giga: prefix_string:= 'G';
        mega: prefix_string:= 'M';
        kilo: prefix_string:= 'k';
        hecto: prefix_string:= 'h';
        deca: prefix_string:= 'da';
        deci: prefix_string:= 'd';
        centi: prefix_string:= 'c';
        milli: prefix_string:= 'm';
        micro: prefix_string:= 'u';
        nano: prefix_string:= 'n';
        pico: prefix_string:= 'p';
        femto: prefix_string:= 'f';
        atto: prefix_string:= 'a';
    END_CASE;
ELSE
    prefix_string:= '';
END_IF;

CASE unit.name OF
        metre: unit_string:= 'm';
        gram: unit_string:= 'g';
```

```
            second: unit_string:= 's';
            ampere: unit_string:= 'A';
            kelvin: unit_string:= 'K';
            mole: unit_string:= 'mol';
            candela: unit_string:= 'cd';
            radian: unit_string:= 'rad';
            steradian: unit_string:= 'sr';
            hertz: unit_string:= 'Hz';
            newton: unit_string:= 'N';
            pascal: unit_string:= 'Pa';
            joule: unit_string:= 'J';
            watt: unit_string:= 'W';
            coulomb: unit_string:= 'C';
            volt: unit_string:= 'V';
            farad: unit_string:= 'F';
            ohm: unit_string:= 'Ohm';
            siemens: unit_string:= 'S';
            weber: unit_string:= 'Wb';
            tesla: unit_string:= 'T';
            henry: unit_string:= 'H';
            degree_Celsius: unit_string:= 'Cel';
            lumen: unit_string:= 'lm';
            lux: unit_string:= 'lx';
            becquerel: unit_string:= 'Bq';
            gray: unit_string:= 'Gy';
            sievert: unit_string:= 'Sv';
    END_CASE;

    RETURN(prefix_string + unit_string);

    END_FUNCTION; -- string_for_SI_unit
        (*
```

F.3.10.9 Функция строчного представления единицы измерения (String_for_**unit**)

Функция **string_for_unit** возвращает строчное представление единицы измерения **unit** (в соответствии с ИСО 10303-41), рассматриваемой как параметр.

Примечание — Функция **string_for_unit** не вызывается кодами языка EXPRESS. Это функция-утилита, позволяющая вычислить строчное представление по атрибуту **structured_representation dic_unit,** если строчное представление **string_representation** отсутствует. Данное строчное представление соответствует варианту, рассмотренному в приложении В МЭК 61360-1:2009.

Пример представления на языке EXPRESS:

```
        *)
        FUNCTION string_for_unit(u: unit): STRING;
            IF 'MEASURE_SCHEMA.DERIVED_Unit' IN TYPEOF(u)
            THEN
                RETURN(string_for_derived_unit(u));
            ELSE (* 'MEASURE_SCHEMA.NAMED_Unit' IN TYPEOF(u) holds true      *)
                RETURN(string_for_named_unit(u));
            END_IF;
        END_FUNCTION; -- string_for_unit
        (*
```

F.3.10.10 Функция обеспечения доступа ко всем описаниям класса (All_class_descriptions_reachable)

Функция **all_class_descriptions_reachable** проверяет тот факт, что если словарные элементы **dictionary_element** дают описание класса, и на них производится ссылка БСЕ некоторого класса **class_BSU** и всех его суперклассов, то они могут быть вычислены в дереве наследственности, определенном иерархией классов.

Пример представления на языке EXPRESS:

```
        *)
        FUNCTION all_class_descriptions_reachable(cl: class_BSU): BOOLEAN;
        IF NOT EXISTS(cl)
        THEN
            RETURN(?);
        END_IF;
        IF SIZEOF(cl.definition) = 0
        THEN
```

```
        RETURN(FALSE);
    END_IF;
    IF NOT(EXISTS(cl.definition[1]\class.its_superclass))
    THEN
        RETURN(TRUE);
    ELSE
        RETURN(all_class_descriptions_reachable(
            cl.definition[1]\class.its_superclass));
    END_IF;
    END_FUNCTION; -- all_class_descriptions_reachable
    (*
```

F.3.10.11 Функция вычисления известных видимых свойств (Compute_known_visible_properties)

Функция **compute_known_visible_properties** вычисляет набор свойств, являющихся видимыми в данном классе. Если определение недоступно, то функция возвращает только те видимые свойства, которые могут быть вычислены.

Примечание — Если словарные определения **dictionary_definition** некоторого класса не присутствуют в рассматриваемом контексте обмена (а контекст обмена библиотеки PLIB никогда не предполагается полным), то суперкласс некоторого класса может быть неизвестен. Следовательно, свойства, определенные как видимые в данном суперклассе, не могут быть вычислены функцией **compute_known_visible_properties**. Только для получающей системы все словарные определения **dictionary_definition** базовых семантических единиц (БСЕ) являются доступными. Следовательно, для получающей системы, функция **compute_known_visible_properties** вычисляет все свойства, являющиеся видимыми в классе, путем ссылки на него (или на любой его суперкласс) с помощью атрибута **name_scope.**

Пример представления на языке EXPRESS:

```
    *)
    FUNCTION compute_known_visible_properties(cl: class_BSU):
        SET OF property_BSU;
    LOCAL
        s: SET OF property_BSU:= [ ];
    END_LOCAL;

    s:= s + USEDIN(cl, 'ISO13584_IEC61360_DICTIONARY_SCHEMA' +
        '.PROPERTY_BSU.NAME_SCOPE');
    IF SIZEOF(cl.definition) = 0
    THEN
        RETURN(s);
    ELSE
        IF EXISTS(cl.definition[1]\class.its_superclass) THEN
            s:= s + compute_known_visible_properties(
                cl.definition[1]\class.its_superclass);
        END_IF;
        RETURN(s);
    END_IF;
    END_FUNCTION; -- compute_known_visible_properties
    (*
```

F.3.10.12 Функция вычисления известного видимого типа данных (Compute_known_visible_**data_type**)

Функция **compute_known_visible_data_type** вычисляет набор типов данных **data_types,** являющихся видимыми в данном классе. Если определение недоступно, то функция возвращает только видимые типы данных **data_types**, которые могут быть вычислены.

Примечание — Если словарные определения **dictionary_definition** некоторых классов не присутствуют в рассматриваемом контексте обмена (а контекст обмена библиотеки PLIB никогда не предполагается полным), то суперкласс некоторого класса может быть неизвестен. Следовательно, типы данных **data_types,** определенные как видимые в данном суперклассе, не могут быть вычислены функцией **compute_known_visible_data_type**. Только для получающей системы все словарные определения **dictionary_definition** базовых семантических элементов (БСЕ) являются доступными. Следовательно, для получающей системы функция **compute_known_visible_data_type** вычисляет все типы данных **data_types,** являющиеся видимыми в классе, путем ссылки на данный класс (или любой его суперкласс) с помощью атрибута **name_scope**.

Пример представления на языке EXPRESS:

```
    *)
    FUNCTION compute_known_visible_data_types(cl: class_BSU):
        SET OF data_type_BSU;
    LOCAL
```

```
        s: SET OF data_type_BSU:=[ ];
    END_LOCAL;

    s:= s + USEDIN(cl, 'ISO13584_IEC61360_DICTIONARY_SCHEMA' +
        '.Data_type_BSU.NAME_SCOPE');
    IF SIZEOF(cl.definition) = 0
    THEN
        RETURN(s);
    ELSE
        IF EXISTS(cl.definition[1]\class.its_superclass)
        THEN
            s:= s + compute_known_visible_data_types(
                cl.definition[1]\class.its_superclass);
        END_IF;
        RETURN(s);
    END_IF;
    END_FUNCTION; -- compute_known_visible_data_types
    (*
```

F.3.10.13 Функция вычисления известных применимых свойств (Compute_known_applicable_properties)

Функция **compute_known_applicable_properties** вычисляет набор свойств, являющихся применимыми в данном классе. Если определение недоступно, то функция возвращает только те применимые свойства, которые могут быть вычислены.

Примечание — Если словарное определение **dictionary_definition** некоторого класса не присутствует в рассматриваемом контексте обмена (а контекст обмена библиотеки PLIB никогда не предполагается полным), то суперкласс некоторого класса может быть неизвестен. Следовательно, свойство, определенное как применимое в данном суперклассе, не может быть вычислено функцией **compute_known_applicable_properties.** Только для получающей системы все словарные определения **dictionary_definition** базового семантического элемента (БСЕ) являются доступными. Следовательно, для получающей системы функция **compute_known_applicable_properties** вычисляет все свойства, являющиеся применимыми в данном классе, так как на них производится ссылка атрибутом **described_by,** или они импортируются с помощью априорного семантического соотношения **a_priori_semantic_relationship.**

Пример представления на языке EXPRESS:

```
    *)
    FUNCTION compute_known_applicable_properties(cl: class_BSU):
        SET OF property_BSU;
    LOCAL
        s: SET OF property_BSU:= [ ];
    END_LOCAL;

    IF SIZEOF(cl.definition)=0
    THEN
        RETURN(s);
    ELSE
        REPEAT i:= 1 TO SIZEOF(cl.definition[1]\class.described_by);
            s:= s + cl.definition[1]\class.described_by[i];
        END_REPEAT;

        IF (('ISO13584_IEC61360_ITEM_CLASS_CASE_OF_SCHEMA.'
            + 'A_PRIORI_SEMANTIC_RELATIONSHIP')
            IN TYPEOF (cl.definition[1]))
        THEN
            s:= s + cl.definition[1]\a_priori_semantic_relationship
                .referenced_properties;
        END_IF;
    IF EXISTS(cl.definition[1]\class.its_superclass)
        THEN
            s:= s + compute_known_applicable_properties(
                cl.definition[1]\class.its_superclass);
        END_IF;

        RETURN(s);
    END_IF;
    END_FUNCTION; -- compute_known_applicable_properties
    (*
```

F.3.10.14 Функция вычисления известного применимого типа данных (Compute_known_applicable_**data_type**s)

Функция **compute_known_applicable_data_types** вычисляет набор типов данных **data_type,** являющихся применимыми в данном классе. Если определение недоступно, то функция возвращает только те применимые типы данных **data_types,** которые могут быть вычислены.

Примечание — Если словарное определение **dictionary_definition** некоторого класса не присутствует в рассматриваемом контексте обмена (а контекст обмена библиотеки PLIB никогда не предполагается полным), то суперкласс некоторого класса может быть неизвестен. Следовательно, типы данных **data_types,** определенные как применимые в данном суперклассе, не могут быть вычислены функцией **compute_known_applicable_data_type**. Только для получающей системы все словарные определения **dictionary_definition** базовых семантических единиц (БСЕ) являются доступными. Следовательно, для получающей системы функция **compute_known_applicable_data_types** вычисляет все типы данных **data_types,** являющиеся применимыми в классе, так как на них производится ссылка атрибутом **defined_type,** или они импортируются с помощью априорного семантического соотношения **a_priori_semantic_relationship.**

Пример представления на языке EXPRESS:

```
*)
FUNCTION compute_known_applicable_data_types(cl: class_BSU):
    SET OF data_type_BSU;
LOCAL
    s: SET OF data_type_BSU:= [ ];
END_LOCAL;

IF SIZEOF(cl.definition) = 0
THEN
    RETURN(s);
ELSE
    REPEAT i:= 1 TO SIZEOF(cl.definition[1]\class.defined_types);
        s:= s + cl.definition[1]\class.defined_types[i];
    END_REPEAT;
    IF (('ISO13584_IEC61360_ITEM_CLASS_CASE_OF_SCHEMA.'
        + 'A_PRIORI_SEMANTIC_RELATIONSHIP')
        IN TYPEOF (cl.definition[1]))
    THEN
        s:= s + cl.definition[1]\a_priori_semantic_relationship
            .referenced_data_types;
    END_IF;

    IF EXISTS(cl.definition[1]\class.its_superclass)
    THEN
        s:= s + compute_known_applicable_data_types(
            cl.definition[1]\class.its_superclass);
    END_IF;

    RETURN(s);
END_IF;
END_FUNCTION; -- compute_known_applicable_data_types
(*
```

F.3.10.15 Функция создания набора из элементов списка (List_to_set)

Функция **list_to_set** создает набор из элементов списка с именем **l**. Тип элементов набора совпадает с типом элементов в исходном списке.

Пример представления на языке EXPRESS:

```
*)
FUNCTION list_to_set(l: LIST [0:?] OF GENERIC:type_elem):
    SET OF GENERIC: type_elem;

LOCAL
    s: SET OF GENERIC: type_elem:= [ ];
END_LOCAL;

REPEAT i:= 1 TO SIZEOF(l);
    s:= s + l[i];
END_REPEAT;
```

```
RETURN(s);
END_FUNCTION; -- list_to_set
(*
```

F.3.10.16 Функция проверки применимости свойств (Check_property_applicability)

Функция **check_property_applicability** проверяет тот факт, что только свойства, не являющиеся применимыми в классе по наследству, могут стать применимыми в данном классе, так как на них производится ссылка атрибутом **described_by**.

Пример представления на языке EXPRESS:

```
*)
FUNCTION check_properties_applicability(cl: class): LOGICAL;
LOCAL
    inter: SET OF property_bsu:= [ ];
END_LOCAL;

IF EXISTS(cl.its_superclass)
THEN
        IF (SIZEOF(cl.its_superclass.definition)=1)
        THEN
            inter:= (list_to_set(cl.described_by) *
                cl.its_superclass.definition[1]\class.
                known_applicable_properties);
        RETURN(inter = [ ] );
    ELSE
        RETURN(UNKNOWN);
    END_IF;
ELSE
    RETURN(TRUE);
END_IF;

END_FUNCTION; -- check_properties_applicability
(*
```

F.3.10.17 Функция проверки применимости типов данных (Check_datatypes_applicability)

Функция **check_datatypes_applicability** проверяет тот факт, что только типы данных не являющиеся применимыми в классе по наследству могут стать применимыми в данном классе, так как на них производится ссылка атрибутом **defined_type**.

Пример представления на языке EXPRESS:

```
*)
FUNCTION check_datatypes_applicability(cl: class): LOGICAL;
LOCAL
    inter: SET OF data_type_bsu:= [ ];
END_LOCAL;

IF EXISTS(cl.its_superclass)
THEN
    IF (SIZEOF(cl.its_superclass.definition) = 1)
    THEN
        inter:= cl.defined_types *
            cl.its_superclass.definition[1]\class.
            known_applicable_data_types;
        RETURN(inter = [ ] );
    ELSE
        RETURN(UNKNOWN);
    END_IF;
ELSE
    RETURN(TRUE);
END_IF;

END_FUNCTION; -- check_datatypes_applicability
(*
```

F.3.10.18 Функция проверки уникальности языка перевода (One_language_per_translation)

Функция **one_language_per_translation** проверяет тот факт, что язык перевода административных данных **administrative_data** является уникальным.

Пример представления на языке EXPRESS:

```
*)
FUNCTION one_language_per_translation (adm: administrative_data)
                                        : LOGICAL;
    LOCAL
        count: INTEGER;
        lang: language_code;
    END_LOCAL;

    REPEAT i:= 1 TO SIZEOF (adm.translation);
        lang:= adm.translation[i].language;
        count:= 0;
        REPEAT j:=1 TO SIZEOF (adm.translation);
            IF lang = adm.translation[j].language
            THEN
                count:= count+1;
            END_IF;
        END_REPEAT;
        IF count >1
        THEN RETURN (FALSE);
        END_IF;
    END_REPEAT;
    RETURN(TRUE);

END_FUNCTION; -- one_language_per_translation
(*
```

F.3.10.19 Функция вычисления целых значений неколичественного типа (Allowed_value_integer_type)

Функция **allowed_value_integer_type** вычисляет набор целых **integer_type** значений, допустимых для атрибута **non_quantitative_int_type.** Если рассматриваемый параметр не определен, то функция возвращает неопределенное значение.

Пример представления на языке EXPRESS:

```
*)
FUNCTION allowed_values_integer_types (nqit: non_quantitative_int_type)
                                : SET OF integer_type;
LOCAL
    s: SET OF integer_type:=[ ];
END_LOCAL;

REPEAT i:=1 TO SIZEOF (nqit.domain.its_values);
    s:= s + nqit.domain.its_values[i].value_code;
END_REPEAT;
RETURN(s);

END_FUNCTION; -- allowed_values_integer_types
(*
```

F.3.10.20 Функция идентификации свойств со значением класса (Is_class_valued_properties)

Функция **is_class_valued_properties** возвращает значение TRUE, если свойство **prop** определено как свойство со значением класса в классе **cl** с помощью атрибута **sub_class_properties** в классе **cl** или в любом его суперклассе. Если словарные определения **dictionary_definition** некоторых классов недоступны (при вычислении всех суперклассов для класса **cl**), то функция возвращает UNKNOWN.

Пример представления на языке EXPRESS:

```
*)
FUNCTION is_class_valued_property(
    prop: property_BSU; cl: class_BSU): LOGICAL;
    IF (SIZEOF(cl.definition) = 0)
    THEN
        RETURN (UNKNOWN);
    ELSE
        IF NOT (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
            +'.ITEM_CLASS') IN TYPEOF(cl.definition[1]))
        THEN
            RETURN (FALSE);
        END_IF;
        IF prop IN cl.definition[1].sub_class_properties
```

```
                THEN RETURN (TRUE);
                END_IF;
                IF NOT EXISTS(cl.definition[1].its_superclass)
                THEN
                (* end of chain reached, didn't meet super so far    *)
                        RETURN(FALSE);
                END_IF;
                RETURN(is_class_valued_property(prop,
                        cl.definition[1].its_superclass));
        END_IF;
    END_FUNCTION; -- is_class_valued_property
    (*
```

F.3.10.21 Функция определения значения свойства (Class_value_assigned)

Функция **class_value_assigned** возвращает набор значений свойства **prop,** назначенного для класса **cl**, с помощью атрибута постоянного значения класса **class_constant_value** в классе **cl** или в любом суперклассе для класса **cl.** Если в различных суперклассах назначены различные значения, то функция возвращает набор всех этих назначенных значений. Если словарные определения **dictionary_definition** некоторых классов недоступны при вычислении всех суперклассов для класса **cl,** то возвращаются только вычисленные значения.

Пример представления на языке EXPRESS:

```
    *)
    FUNCTION class_value_assigned (prop: property_BSU;
        cl: class_BSU): SET OF primitive_value;
        LOCAL
            val:SET OF primitive_value:=[ ];
            cva: SET OF class_value_assignment:=[ ];
        END_LOCAL;
        IF (SIZEOF(cl.definition) = 0)
        THEN
            RETURN (val);
        END_IF;
        IF NOT (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
            +'.ITEM_CLASS') IN TYPEOF(cl.definition[1]))
        THEN
            RETURN (val);
        END_IF;
        IF EXISTS(cl.definition[1])
        THEN
            cva:= QUERY
                (a <* cl.definition[1].class_constant_values
                | a.super_class_defined_property = prop);
            REPEAT i:=1 TO SIZEOF (cva);
                val:= val + cva[i].assigned_value;
            END_REPEAT;
            IF NOT EXISTS(cl.definition[1].its_superclass)
            THEN
                RETURN (val);
            ELSE RETURN (val + class_value_assigned
                (prop,cl.definition[1].its_superclass));
            END_IF;
        END_IF;
    END_FUNCTION; -- class_value_assigned

    END_SCHEMA; — ISO13584_IEC61360_dictionary_schema
    (*
```

F.4 Стандартная схема языкового ресурса (ISO13584_IEC61360_language_resource_schema)

Нижеследующая схема обеспечивает ресурсы для представления строк на различных языках. Данная языковая схема извлечена из имеющейся словарной схемы и может быть использована в других схематиках. В значительной степени она основана на схеме поддержки ресурсов **support_resource_schema** из ИСО 10303-41 и может рассматриваться как ее расширение. По данной схеме только один особый язык можно использовать в рассматриваемом контексте обмена (в физическом файле) без дополнительных усложнений, которые становятся необходимыми при использовании нескольких языков. См. графические пояснения на рисунке F.13.

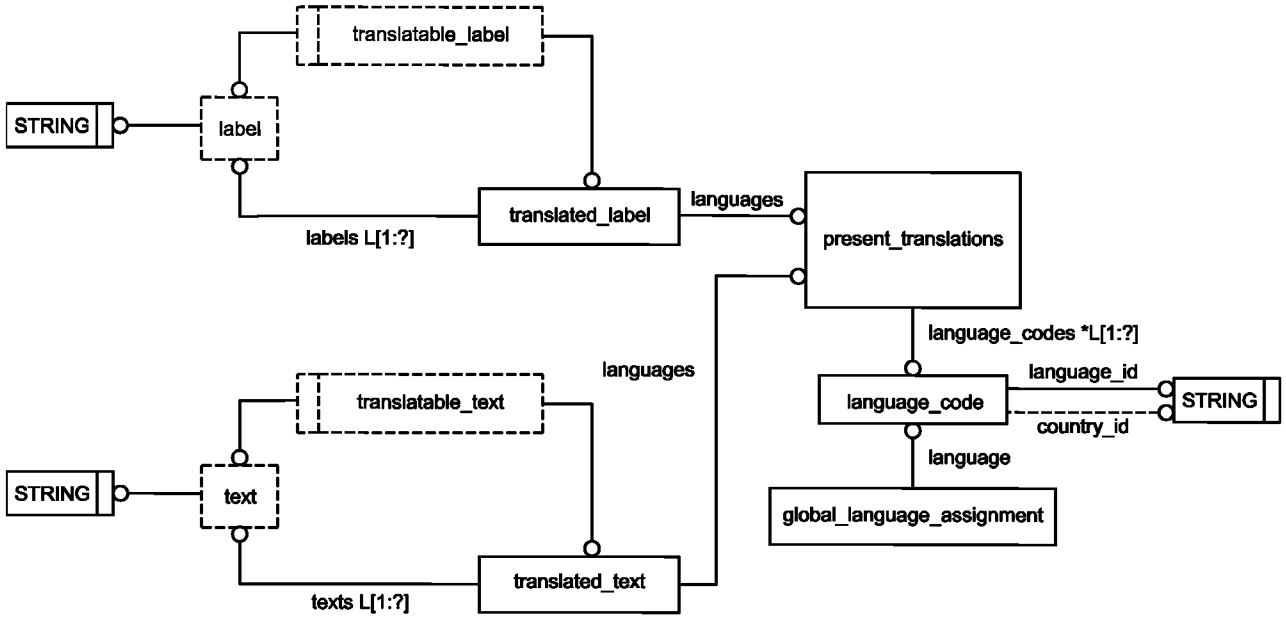


Рисунок 13 — Стандартная схема языкового ресурса ISO13584_IEC61360_language_resource_schema и схема ресурса поддержки support_resource_schema

translatable_label	Переводимая метка
string	Строка
label	Метка
labels L[1:?]	Массив меток
translated_label	Переведенная метка
languages	Языки
present_translations	Доступные переводы
language_codes *L[1:?]	Массив языковых кодов
translatable_text	Переводимый текст
text	Текст
texts L[1:?]	Массив текстов
translated_text	Переведенный текст
language_id	Идентификатор языка
language_code	Код языка
country_id	Идентификатор страны
global_language_assignment	Назначение глобального языка

Пример представления на языке EXPRESS:

```
*)
SCHEMA ISO13584_IEC61360_language_resource_schema;
REFERENCE FROM support_resource_schema(identifier, label, text);
(*
```

Примечание — Вышеуказанная схема ресурса поддержки **support_resource_schema** описана в ИСО 10303-41.

F.4.1 Определения типов и сущностей стандартной схемы языкового ресурса ISO13584_IEC61360_language_resource_schema

Данный подраздел содержит определения типов и сущностей языка EXPRESS для стандартной языковой схемы **ISO13584_IEC61360_language_resource_schema.**

F.4.1.1 Код языка (Language_code)

Сущность **language_code** дает возможность идентифицировать язык в соответствии с ИСО 639-1. Существует два кода языка:

- код языка в соответствии с ИСО 639-1 и ИСО 639-2, а также по выбору
- код страны в соответствии с ИСО 3166-1, указывающий, в какой стране на данном языке говорят.

Пример представления на языке EXPRESS:

```
*)
ENTITY language_code;
    language_id: identifier;
    country_id: OPTIONAL identifier;
WHERE
    WR1: (LENGTH (language_id) = 2) OR (LENGTH (language_id) = 3);
    WR2: LENGTH (country_id) = 2;
END_ENTITY; -- language_code
(*
```

Определения атрибутов:

language_id: код, указывающий язык в соответствии с ИСО 639-1 и ИСО 639-2.

country_id: код, указывающий страну, где на данном языке говорят, в соответствии с ИСО 3166-1.

Пояснения к тексту программы:

WR1: длина кода **language_id** равна 2 или 3.

WR2: длина кода **country_id** равна 2.

F.4.1.2 Назначение глобального языка (**Global_language_assignment**)

Сущность **global_language_assignment** указывает язык для переводимой метки **translatable_label** и переводимого текста **translatable_text,** если и метка, и текст выбраны соответственно (т. е. без явного указания языка, как это сделано в атрибутах переведенная метка **translated_label** и переведенный текст **translated_text).**

Пример представления на языке EXPRESS:

```
*)
ENTITY global_language_assignment;
    language: language_code;
END_ENTITY; -- global_language_assignment
(*
```

Определения атрибутов:

language: код назначенного языка.

F.4.1.3 Доступные переводы (Present_translations)

Сущность **present_translations** служит для указания языков, использованных для получения переведенной метки **translated_label** и переведенного текста **translated_text.**

Пример представления на языке EXPRESS:

```
*)
ENTITY present_translations;
    language_codes: LIST [1:?] OF UNIQUE language_code;
UNIQUE
    UR1: language_codes;
END_ENTITY; -- present_translations
(*
```

Определения атрибутов:

language_codes: список уникальных кодов, соответствующих языкам, на которых сделан перевод.

Пояснение к тексту программы:

UR1: для каждого списка кодов языков **language_codes** существует уникальная реализация доступных переводов **present_translations.**

F.4.1.4 Переводимая метка (Translatable_label)

Сущность **translatable_label** определяет тип значения, которое может быть меткой или переведенной меткой **translated_labels**.

Пример представления на языке EXPRESS:

```
*)
TYPE translatable_label = SELECT(label, translated_label);
END_TYPE; -- translatable_label
(*
```

F.4.1.5 Переведенная метка (Translated_label)

Сущность **translated_label** определяет метки, которые уже переведены, и соответствующий язык перевода.

Пример представления на языке EXPRESS:

```
*)
ENTITY translated_label;
    labels: LIST [1:?] OF label;
    languages: present_translations;
WHERE
    WR1: SIZEOF(labels) = SIZEOF(languages.language_codes);
END_ENTITY; -- translated_label
(*
```

Определения атрибутов:

labels: список меток **labels,** которые уже переведены.

language: список языков, на которые переведены рассматриваемые метки.

Пояснения к тексту программы:

WR1: количество меток **labels,** содержащихся в списке **labels,** равно количеству языков, определяемому в атрибуте **languages.language_codes**.

Дополнительные пояснения:

IP1: содержание массива **labels[i]** на языке, идентифицированном массивом атрибутов **languages.language_codes[i].**

F.4.1.6 Переводимый текст (Translatable_text)

Сущность **translatable_text** определяет тип значений, который может быть текстом или переведенным текстом.

Пример представления на языке EXPRESS:

```
*)
TYPE translatable_text = SELECT(text, translated_text);
END_TYPE; -- translatable_text
(*
```

F.4.1.7 Переведенный текст (Translated_text)

Сущность **translated_text** определяет переведенные тексты и соответствующие языки перевода.

Пример представления на языке EXPRESS:

```
*)
ENTITY translated_text;
    texts: LIST [1:?] OF text;
    languages: present_translations;
WHERE
    WR1: SIZEOF(texts) = SIZEOF(languages.language_codes);
END_ENTITY; -- translated_text
(*
```

Определения атрибутов:

texts: список переведенных текстов.

languages: список языков, на которые переведен каждый текст.

Пояснения к тексту программы:

WR1: количество текстов, содержащихся в списке текстов, равно количеству языков, определенных атрибутом **languages.language_codes**.

Дополнительные пояснения:

IP1: содержание массивов **texts[i]** на языке, идентифицированном атрибутом **languages.language_codes[i].**

F.4.2 Определения функций стандартной схемы языкового ресурса ISO13584_IEC61360_language_resource_schema

Данный подраздел содержит описание функции, на которую производится ссылка в разделах по месту для подтверждения непротиворечивости данных.

F.4.2.1 Функция проверки длины метки (Check_label_length)

Функция **check_label_length** проверяет тот факт, что ни одна из меток массива **I** не превышает длину, указанную атрибутом **l_length.**

Пример представления на языке EXPRESS:

```
*)
FUNCTION check_label_length(I: translatable_label;
    l_length: INTEGER): BOOLEAN;
IF 'ISO13584_IEC61360_LANGUAGE_RESOURCE_SCHEMA.TRANSLATED_LABEL'
    IN TYPEOF(I)
THEN
    REPEAT i:=1 TO SIZEOF(I.labels);
        IF LENGTH(I.labels[i]) > l_length
        THEN
            RETURN(FALSE);
        END_IF;
        END_REPEAT;
```

```
            RETURN(TRUE);

    ELSE (* the argument l is a single string *)
        RETURN(LENGTH(l) <= l_length);
    END_IF;
    END_FUNCTION; -- check_label_length
    (*
```

F.4.3 Определение правила стандартной схемы языкового ресурса ISO13584_IEC61360_language_resource_schema

Правило **single_language_assignment** подтверждает, что только один язык может быть назначен для использования в переводимых метках **translatable_label** и переводимых текстах **translatable_text.**

Пример представления на языке EXPRESS:

```
    *)
    RULE single_language_assignment FOR(global_language_assignment);
    WHERE
        SIZEOF(global_language_assignment) <= 1;
    END_RULE; — single_language_assignment
    END_SCHEMA; — ISO13584_IEC61360_language_resource_schema
    (*
```

F.5 Стандартная схема ограничений класса ISO13584_IEC61360_class_constraint_schema

Данный Раздел определяет требования к схеме ограничений класса **class_constraint_schema.** Нижеследующее объявление языка EXPRESS представляет блок **ISO13584_IEC61360_class_constraint_schema** и задает необходимые внешние ссылки.

Пример представления на языке EXPRESS:

```
    *)
    SCHEMA ISO13584_IEC61360_class_constraint_schema;

    REFERENCE FROM ISO13584_IEC61360_dictionary_schema (
        class_BSU,
        property_BSU,
        definition_available_implies,
        is_subclass,
        data_type,
        simple_type,
        complex_type,
        named_type,
        allowed_values_integer_types);
    REFERENCE FROM ISO13584_extended_dictionary_schema
        data_type_typeof,
        data_type_class_of,
        data_type_type_name);
    REFERENCE FROM ISO13584_instance_resource_schema
        (Boolean_value,
        compatible_class_and_class,
        complex_value,
        dic_class_instance,
        entity_instance_value,
        int_level_spec_value,
        integer_value,
        level_spec_value,
        number_value,
        primitive_value,
        rational_value,
        real_level_spec_value,
        real_value,
        right_values_for_level_spec,
        same_translations,
        simple_value,
        string_value,
        translatable_string_value,
        translated_string_value,
        property_or_data_type_BSU);
    REFERENCE FROM ISO13584_aggregate_value_schema
```

```
        (aggregate_entity_instance_value,
        list_value,
        set_value,
        bag_value,
        array_value,
        set_with_subset_constraint_value,
        compatible_complete_types_and_value);
    (*
```

Примечание — Схематики, на которые выше производятся ссылки, описаны в нижеследующих документах:

ISO13584_IEC61360_dictionary_schema в МЭК 61360-2
(дублируется для удобства в разделе 4.5 и далее).
ISO13584_extended_dictionary_schema в ИСО 13584-24:2003
ISO13584_instance_resource_schema в ИСО 13584-24:2003
ISO13584_aggregate_value_schema в ИСО 13584-25

F.5.1 Введение в стандартную схему ограничений класса ISO13584_IEC61360_class_constraint_schema

Схема **ISO13584_IEC61360_class_constraint_schema** задает конструктивы языка EXPRESS, позволяющие переопределить, путем задания ограничений, область значений данного свойства, если оно применяется в подклассе характеристического класса, где данное свойств определено как видимое. Данное ограничение должно явно описывать только те ограничения области значений, которые следуют из структуры класса.

Пример — В ИСО 13584-511 класс болтов/винтов с метрической резьбой определен следующим образом: «крепежный элемент с головкой, наружной резьбой, цилиндрическим телом, на котором резьба нарезана частично или полностью, и головкой с приспособлениями для завинчивания». Данный класс имеет, среди прочих, два свойства: тип головки и свойства головки. Область значений свойства тип головки — это неколичественный тип данных, включающий, главным образом, нижеследующие значения: шестигранная_головка, восьмигранная_головка и круглая_головка. Свойство свойства головки является особенностью. Это означает, что она имеет тип данных item_class. Областью значений является класс головка, определяющий любой вид головки. Класс головка имеет несколько подклассов: шестигранная головка, ассоциированная со всеми свойствами, позволяющими описывать шестигранную головку (например, размер под ключ), и круглая головка, ассоциированная со всеми свойствами, позволяющими описывать круглую головку (например, диаметр головки).

Класс *болтов/винтов с метрической резьбой* имеет подкласс, называемый *винты с шестигранной головкой* и определяемый следующим образом: «крепежный элемент с метрической наружной резьбой, шестигранной головкой и резьбой, нарезанной до головки». Данный класс унаследует свойства: *тип головки* и *свойства головки.* Из определения подкласса *винтов с шестигранной головкой* ясно, что свойство *тип головки* может принимать только значение *шестигранная_головка.* Свойство *свойства головки* может быть только реализацией класса особенностей *шестигранная головка.* Однако указанные ограничения являются неявными: они просто фиксируются неформально в определении сущности.

Таким образом, указанные ограничения не являются компьютерными. Рассматриваемые ограничения, определенные стандартной схемой ограничений класса **ISO13584_IEC61360_class_constraint_schema,** позволяют сделать нижеследующие два ограничения явными путем ассоциирования их с классом *винтов с шестигранной головкой*: (1) ограничение нумерации **enumeration_constraint** для свойства *тип головки* (допускающее только код *шестигранная_головка*) и (2) **subclass_constraint** для свойства *свойства головки* (допускающее только класс особенностей *шестигранная головка*).

Ограничения наследуются. Если свойство, область значений которого имеет одно ограничение в классе С, требует задания другого ограничения в подклассе С, то оба ограничения применяются совместно. Таким образом. реальная область значений в подклассе С — это пересечение двух областей, определенных двумя ограничениями. Предложенный механизм аналогичен механизму повторного определения типа на языке EXPRESS.

Данная схема позволяет выразить ограничения, применимые к типам данных системы типов стандартной словарной схемы **ISO13584_IEC61360_dictionary_schema.** По правилу для данных сущностей, ссылающихся на ограничение, каждое ограничение применимо к типу данных, к которому оно относится.

F.5.2 Определения сущностей стандартной схемы ограничений класса ISO13584_IEC61360_class_constraint_schema

Данный Раздел определяет сущности стандартной схемы ограничений класса **ISO13584_IEC61360_class_constraint_schema**.

F.5.2.1 Ограничение (Constraint)

Сущность **constraint** позволяет дать определение ограничения.

Пример представления на языке EXPRESS:

```
    *)
    ENTITY constraint
    ABSTRACT SUPERTYPE OF (ONEOF (
                property_constraint,
                class_constraint));
        constraint_id: OPTIONAL constraint_identifier;
    END_ENTITY; -- constraint
    (*
```

Определения атрибутов:

constraint_id: атрибут **constraint_identifier,** задающий ограничения.

F.5.2.2 Ограничение свойства (Property_constraint)

Сущность **property_constraint** — это ограничение набора реализаций класса путем задания одного условия для области значений некоторого свойства данного класса.

Пример представления на языке EXPRESS:

```
*)
ENTITY property_constraint
ABSTRACT SUPERTYPE OF (ONEOF (
        integrity_constraint,
        context_restriction_constraint))
SUBTYPE OF (constraint);
    constrained_property: property_BSU;
END_ENTITY; -- property_constraint
(*
```

Определения атрибутов:

constrained_properties: базовая семантическая единица свойства **property_BSU,** для которой применяется ограничение.

F.5.2.3 Ограничение класса (Class_constraint)

Сущность **class_constraint** — это ограничение, накладывающее условие на допустимый набор реализаций класса. При этом рассматриваются несколько свойств или глобальных ограничений.

Пример представления на языке EXPRESS:

```
*)
ENTITY class_constraint
ABSTRACT SUPERTYPE OF (configuration_control_constraint)
SUBTYPE OF (constraint);
END_ENTITY; -- class_constraint
(*
```

F.5.2.4 Ограничение управления конфигурацией (Configuration_control_constraint)

Сущность **configuration_control_constraint** накладывает условие на набор реализаций, на которые производится ссылка. На эти реализации некоторая другая ссылающаяся реализация может ссылаться прямо или косвенно с помощью цепочки свойств. Ссылающаяся реализация — это любая реализация класса, ссылающаяся на ограничение управления конфигурацией **configuration_control_constraint** с помощью атрибута **constraint** или наследующая его в классе, обладающем данным свойством. Сущность **configuration_control_constraint** определяет вспомогательное предварительное условие **precondition** наложения ограничения на ссылающуюся реализацию. Она определяет также выходное условие **postcondition,** задающее допустимые наборы значений некоторых свойств реализаций класса, на которые производится ссылка.

Пример — Болтовое соединение состоит из нижеследующего набора крепежных элементов: один крепежный элемент с наружной резьбой, некоторое количество шайб и одна или несколько гаек. Существуют различные виды резьб, включая резьбу винта-самореза, резьбу винта по дереву, метрическую наружную резьбу, метрическую внутреннюю резьбу, дюймовую внутреннюю резьбу и дюймовую наружную резьбу. Предположим, что нужно дать описание метрического болтового соединения. Нужно гарантировать, что какая бы ни была точностная структура соединения, и крепежный элемент с наружной резьбой и все гайки данного соединения имеют метрическую резьбу. Для этого в классе метрических болтовых соединений задается ограничение управления конфигурацией configuration_control_constraint, гарантирующее, что любой крепежный элемент (соответствующий Стандарту ISO 13584-511), на который производится ссылка некоторой реализацией данного класса (или любого его подкласса), либо должен принадлежать классам, в которых не задано значение свойства тип резьбы (например, шайба), либо должен принадлежать классам, значения которых принадлежат паре {метрическая наружная резьба, метрическая внутренняя резьба}.

Примечание 1 — Как предварительное условие **precondition**, так и выходное условие **postcondition** могут ограничивать только те свойства, данные которых имеют неколичественный кодовый тип **non_quantitative_code_type**. Такие свойства могут быть заданы для значений либо на уровне реализации, либо на уровне класса, если они объявлены как свойства со значением класса, т. е. свойства подкласса **sub_class_properties** в некотором классе.

Примечание 2 — Свойства, на которые производится ссылка в предварительном условии **precondition**, должны быть применимы в классе, ссылающемся на ограничение управления конфигурацией **configuration_control_constraint**.

Примечание 3 — В ограничении управления конфигурацией **configuration_control_constraint** используются фильтры **filters**. Они накладывают предварительное условие на ссылающуюся реализацию и задают ограничения на реализации, на которые производится ссылка.

Пример представления на языке EXPRESS:

```
*)
ENTITY configuration_control_constraint
SUBTYPE OF (class_constraint);
        precondition: SET [0:?] OF filter;
        postcondition: SET [1:?] OF filter;
END_ENTITY; -- configuration_control_constraint
(*
```

Определения атрибутов:

precondition: фильтры **filters** ссылающейся реализации для наложения ограничения.

Примечание 4 — Если набор фильтров пуст, то ограничение накладывается на любую ссылающуюся реализацию.

postcondition: фильтры **filters** реализации, на которую ссылаются, для получения разрешения на данную ссылку.

F.5.2.5 Фильтр (Filter)

Сущность **filter** — это ограничение нумерации **enumeration_constraint,** ограничивающее допустимую область свойства, тип данных которого либо неколичественный кодовый **non_quantitative_code_type,** либо неколичественный целый **non_quantitative_int_type.**

Пример представления на языке EXPRESS:

```
*)
ENTITY filter;
        referenced_property: property_BSU;
        domain: enumeration_constraint;
WHERE
        WR1: definition_available_implies (
                referenced_property,
                (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
                +'.NON_QUANTITATIVE_CODE_TYPE') IN TYPEOF(
                referenced_property.
                definition[1]\property_DET.domain))
                OR
                (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
                +'.NON_QUANTITATIVE_INT_TYPE') IN TYPEOF(
                referenced_property.
                definition[1]\property_DET.domain)));
        WR2: definition_available_implies (
                referenced_property,
                correct_constraint_type(domain,
                referenced_property.definition[1].domain));
END_ENTITY; -- filter
(*
```

Определения атрибутов:

referenced_properties: свойство, область значений которого ограничена фильтром **filter.**

domain: ограничение нумерации **enumeration_constraint,** ограничивающее область значений свойства, на которое производится ссылка.

Пояснения к тексту программы:

WR1: тип данных ссылочного свойства **referenced_property** — это либо неколичественный кодовый **non_quantitative_code_type,** либо неколичественный целый **non_quantitative_int_type.**

WR2: сущность **domain** должна определять область значений, которая может ограничивать исходную область значений свойства.

F.5.2.6 Ограничение целостности (Integrity_constraint)

Сущность **integrity_constraint** — это особое ограничение свойства, которое объявляет явно, что для некоторого конкретного класса (как результат определения данного класса и всех его подклассов) только ограничение области значений, идентифицированное типом данных, допустимо для свойства.

Пример — В ссылочном словаре, определенном для крепежных элементовв ИСО 13584-511, болт/винт с метрической резьбой имеет свойство свойства головки, которое может приобретать, в качестве значения, член любого подкласса рассматриваемого класса особенностей головки. Если данный болт/винт с метрической резьбой также является членом подкласса винтов с шестигранной головкой, то свойство свойства головки может только быть членом класса особенностей шестигранной головки. В противном случае рассматриваемый болт/винт с метрической резьбой не может быть членом подкласса винтов с шестигранной головкой.

Примечание — В рассмотренном выше примере ограничение целостности никак не изменяет смысл свойства *свойства головки,* унаследованное от *болта/винта с метрической резьбой* в *винт с шестигранной головкой.* Рассматриваемое ограничение явно указывает тот факт, что в контексте подкласса *винтов с шестигранной головкой* допустимым остается только подмножество значений, допустимых для данного свойства в контексте класса *болтов/винтов с метрической резьбой*.

Пример представления на языке EXPRESS:

```
*)
ENTITY integrity_constraint
SUBTYPE OF (property_constraint);
    redefined_domain: domain_constraint;
WHERE
    WR1: definition_available_implies (constrained_property,
        correct_constraint_type(redefined_domain,
        constrained_property.definition[1].domain));
END_ENTITY; -- integrity_constraint
(*
```

Определения атрибутов:

redefined_domain: ограничение, накладываемое на область значений ограниченного свойства.

Пояснения к тексту программы:

WR1: повторно определенная область **redefined_domain** должна задавать область значений, ограничивающую исходную область значений свойства.

F.5.2.7 Ограничение на условия контекста (Context_restriction_constraint)

Сущность **context_restriction_constraint** — это ограничение свойства **property_constraint,** накладывающее условие на допустимую область значений контекстных параметров, от которых зависит рассматриваемое контекстно-зависимое свойство.

Пример представления на языке EXPRESS:

```
*)
ENTITY context_restriction_constraint
SUBTYPE OF (property_constraint);
    context_parameter_constraints: SET [1:?] OF property_constraint;
WHERE
    WR1: definition_available_implies(constrained_property,
        QUERY (cp <*SELF.context_parameter_constraints
        | NOT (cp.constrained_property IN
        constrained_property.definition[1].depends_on))=[ ]);
    WR2: QUERY (cp <*SELF.context_parameter_constraints
        | NOT (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
        +'.INTEGRITY_CONSTRAINT') IN TYPEOF (cp))) =[ ];
    WR3: definition_available_implies(constrained_property,
        'ISO13584_IEC61360_DICTIONARY_SCHEMA.DEPENDENT_P_DET'
        IN TYPEOF(constrained_property.definition[1]));
END_ENTITY; -- context_restriction_constraint
(*
```

Определения атрибутов:

context_parameter_constraints: ограничение, применимое в области значений контекстных параметров.

Пояснения к тексту программы:

WR1: набор свойств, область значений которого ограничена свойством **context_parameter_constraint,** должен являться набором контекстных параметров, от которых зависит рассматриваемое ограниченное свойство.

WR2: все ограничения контекстных параметров **context_parameter_constraint** должны быть ограничениями целостности **integrity_constraints.**

WR3: ограниченное свойство **constrained_property** должно быть контекстно-зависимым типом данных **property dependent_P_DET.**

F.5.2.8 Ограничение области (Domain_constraint)

Сущность **domain_constraint** задает условие, ограничивающее область значений типа данных.

Пример представления на языке EXPRESS:

```
*)
ENTITY domain_constraint
ABSTRACT SUPERTYPE OF(ONEOF(
    subclass_constraint,
    entity_subtype_constraint,
    enumeration_constraint,
    range_constraint,
    string_size_constraint,
```

```
        string_pattern_constraint,
        cardinality_constraint
        ));
    END_ENTITY; -- domain_constraint
    (*
```

F.5.2.9 Ограничение подкласса (Subclass_constraint)

Сущность **subclass_constraint** ограничивает область значений типа ссылки на класс **class_reference_type** для одного или нескольких подклассов рассматриваемого класса, определяющего исходную область.

Пример представления на языке EXPRESS:

```
    *)
    ENTITY subclass_constraint
    SUBTYPE OF(domain_constraint);
        subclasses: SET [1:?] OF class_BSU;
    END_ENTITY; -- subclass_constraint
    (*
```

Определения атрибутов:

subclasses: базовые семантические единицы класса **class_BSUs,** переопределяющие данный класс, которому должны принадлежать значения ограниченного свойства **constrained_property**.

F.5.2.10 Ограничение подтипа сущности (Entity_subtype_constraint)

Сущность **entity_subtype_constraint** ограничивает область значений типа реализации сущности **entity_instance_type** до подтипа СУЩНОСТИ, определяющей ее исходную область.

Пример представления на языке EXPRESS:

```
    *)
    ENTITY entity_subtype_constraint
    SUBTYPE OF(domain_constraint);
        subtype_names: SET[1:?] OF STRING;
    END_ENTITY; -- entity_subtype_constraint
    (*
```

Определения атрибутов:

subtype_names: набор строк, описывающих (в формате функции типа TYPEOF языка EXPRESS) имена типов данных сущности языка EXPRESS, принадлежащих результату действия функции типа TYPEOF языка EXPRESS, примененной к значению, ссылающемуся на повторно определенное ограниченное свойство **constrained_properties**.

F.5.2.11 Ограничение нумерации (Enumeration_constraint)

Сущность **enumeration_constraint** ограничивает область значений типа данных до списка значений, определенных в расширении. Порядок, определенный списком, — это рекомендуемый порядок представления информации. Конкретное описание по желанию может быть ассоциировано с каждым значением подмножества с помощью неколичественного целого типа **non_quantitative_int_type,** *i*-е значение которого описывает смысл *i*-го значения подмножества.

Для подтипов **number_type,** ассоциированных со словарными единицами измерения **dic_unit** и альтернативными единицами измерения и, возможно, с идентификаторами словарных единиц измерения **dic_unit_identifier,** а также с идентификаторами альтернативных единиц измерения, данное ограничение применяется к значению, соответствующему словарной единице **dic_unit** или одному идентификатору словарной единицы **dic_unit_identifier.** Если существуют и сама единица, и ее идентификатор, то они соответствуют одной и той же единице измерения.

Для подтипов **number_type,** ассоциированны с валютой, ограничения накладываются на валюту, указанную в определении их типов данных. Если никакая валюта в определении типов данных не указана, то рассматриваемое ограничение не используется.

Для значений, принадлежащих переводимому строчному типу **translatable_string_type,** ограничения накладываются на строку, представленную на исходном языке, на котором определена область значений свойства. Данный исходный язык может быть определен атрибутом **source_language** административных данных **administrative_data** свойства. Если данный атрибут не существует, то указанный исходный язык считается известным пользователю словаря.

Если какое-либо другое ограничение нумерации **enumeration_constraint** накладывается на свойство ранее уже ассоциированное с некоторым ограничением нумерации **enumeration_constraint** в некотором суперклассе, то применяются оба ограничения. Таким образом, допустимый набор значений — это пересечение двух подмножеств. Порядок представления информации и возможный смысл ограничения, ассоциированного с каждым значением для рассматриваемого ограничения нумерации **enumeration_constraint,** иллюстрируется примерами ниже.

Пример 1 — Если (в классе С1) данное свойство ассоциировано с ограничением нумерации enumeration_constraint, атрибут subset которого равен {1, 3, 5, 7}, то в классе С1 и в любом его подклассе свойство Р1 может принимать только одно из четырех нижеследующих значений: 1, 3, 5, 7.

Пример 2 — Если тип данных свойства Р1 — это целочисленный массив LIST [1:4], и если в классе С1 данное свойство ассоциировано с ограничением нумерации enumeration_constraint, атрибут subset которого { {1}, {3, 5}, {7}, {1, 3, 7}}, то в классе С1 и в любом его подклассе свойство Р1 может принимать только одно из четырех нижеследующих значений: {1}, {3, 5}, {7}, {1, 3, 7}.

Пример представления на языке EXPRESS:

```
*)
ENTITY enumeration_constraint
SUBTYPE OF (domain_constraint);
        subset: LIST [1:?] OF UNIQUE primitive_value;
        value_meaning: OPTIONAL non_quantitative_int_type;
WHERE
        WR1: (NOT(EXISTS(SELF.value_meaning)))
                OR
                (integer_values_in_range(1, SIZEOF(SELF.subset))
                = allowed_values_integer_types(SELF.value_meaning));
END_ENTITY; -- enumeration_constraint
(*
```

Определения атрибутов:

subset: список, описывающий подмножество значений, допустимых для свойства, идентифицированного атрибутом **constrained_properties.**

value_смысл: набор словарных значений **dic_value,** определяющих смысл каждого значения, принадлежащего списку **subset.**

Пояснения к тексту программы:

WR1: если существует неколичественный целый тип **non_quantitative_int_type** для смысла значения **value_смысл**, то набор кодов **value_codes**

его словарных значений **dic_value** должен принадлежать подмножеству 1.. SIZE_OF.

F.5.2.12 Ограничение диапазона (Range_constraint)

Сущность **range_constraint** ограничивает область значений упорядоченного типа подмножеством значений, определенных некоторым диапазоном.

Примечание 1 — Строка не рассматривается как упорядоченный тип и не может быть ограничена условием **range_constraint**.

Для подтипов **number_type,** ассоциированных со словарными единицами измерения **dic_unit,** альтернативными единицами измерения, а также, возможно, с идентификаторами словарных единиц измерения **dic_unit_identifier** и идентификаторами альтернативных единиц измерения, рассматриваемое ограничение накладывается на значение, соответствующее словарной единице **dic_unit** или одному инедтификатору словарной единицы **dic_unit_identifier.** Если существуют и словарная единица, и идентификатор, то они должны соответствовать одной и той же единице измерения.

Для подтипов **number_type,** ассоциированных с валютой, ограничение накладывается на валюту, указанную в определении их типов данных. Если никакая валюта в определении типа данных не указана, то ограничение не используется.

Примечание 2 — Для неколичественного целого типа данных **non_quantitative_int_type** ограничения накладываются на код значения **value_code**.

Пример представления на языке EXPRESS:

```
*)
ENTITY range_constraint
SUBTYPE OF (domain_constraint);
        min_value, max_value: OPTIONAL NUMBER;
        min_inclusive, max_inclusive: OPTIONAL BOOLEAN;
WHERE
        WR1: min_value <= max_value;
        WR2: TYPEOF(min_value) = TYPEOF(max_value);
        WR3: NOT EXISTS (min_value) OR EXISTS (min_inclusive);
        WR4: NOT EXISTS (max_value) OR EXISTS (max_inclusive);
END_ENTITY; -- range_constraint
(*
```

Определения атрибутов:

min_value: число, определяющее нижнюю границу диапазона значений; если это число не задано, то нижней границы нет.

max_value: число, определяющее верхнюю границу диапазона значений; если это число не задано, то верхней границы нет.

min_inclusive: указывает, действительно ли минимальное значение **min_value** принадлежит рассматриваемому диапазону; если такое значение не задано, то нижней границы нет.

max_inclusive: указывает действительно ли максимальное значение **max_value** принадлежит рассматриваемому диапазону; если такое значение не задано, то верхней границы нет.

Пояснения к тексту программы:

WR1: min_value должно быть меньше или равно **max_value.**

WR2: min_value и **max_value** должны иметь один и тот же тип данных.
WR3: если **min_value** задано, то **min_inclusive** также должно быть задано.
WR4: если **max_value** задано, то **max_inclusive** также должно быть задано.
F.5.2.13 Ограничение на длину строки (String_size_constraint)
Сущность **string_size_constraint** ограничивает длину строки, допустимую типом строки или ее подтипом.

Примечание 1 — Областью значений свойства **string_type** может быть строчный тип **string_type**, непереводимый строчный тип **non_translatable_string_type**, переводимый строчный тип **translatable_string_type**, тип универсального идентификатора ресурса **URI_type**, неколичественный кодовый тип **non_quantitative_code_type**, тип данных о дате **date_data_type**, тип данных о времени **time_data_type** или тип данных о дате и времени **date_time_data_type**.

Примечание 2 — Для неколичественного кодового типа **non_quantitative_code_type** ограничение накладывается на код.

Для значений, принадлежащих переводимому строчному типу **translatable_string_type,** ограничение накладывается на строку, представленную на исходном языке, на котором дано определение области значений свойств. Исходный язык может быть определен атрибутом **source_language** административных данных **administrative_data** свойства. Если данный атрибут не существует, то рассматриваемый исходный язык предполагается известным пользователю словаря.

Пример представления на языке EXPRESS:

```
*)
ENTITY string_size_constraint
SUBTYPE OF (domain_constraint);
        min_length: OPTIONAL INTEGER;
        max_length: OPTIONAL INTEGER;
WHERE
        WR1: (min_length >= 0) AND (max_length >= min_length);
END_ENTITY; -- string_size_constraint
(*
```

Определения атрибутов:

min_length: минимальная длина строки, допустимая в качестве значения свойства, идентифицированного атрибутом **constrained_properties**.

max_length: максимальная длина строки, допустимая в качестве значения свойства, идентифицированного атрибутом **constrained_properties**.

Примечание 3 — Если значение **min_length** не существует, то берется 0. Если значение **max_length** не существует, то берется бесконечность.

Пояснения к тексту программы:

WR1: значения **min_length** и **max_length** задают корректные границы.

F.5.2.14 Ограничение на шаблон строки (String_pattern_constraint)

Сущность **string_pattern_constraint** ограничивает область значений строчного типа **string_type** или любого его подтипа до значений, заданных конкретным шаблоном. Синтаксис шаблона определен регулярным выражением языка XML и ассоциированными алгоритмами сравнения, определенными в Схеме XML Часть 2: «Рекомендованные типы данных».

Примечание 1 — Область значений свойства **string_type** — это строчный тип **string_type**, непереводимый строчный тип non_**translatable_string_type**, переводимый строчный тип **translatable_string_type**, тип универсального идентификатора ресурса **URI_type**, неколичественный кодовый тип **non_quantitative_code_type**, тип данных даты date_**data_type**, тип данных времени **time_data_type** или тип данных даты и времени **date_time_data_type**.

Для строчного типа **string_type**, непереводимого строчного типа **non_translatable_string_type**, типа универсального идентификатора ресурса **URI_type**, неколичественного кодового типа **non_quantitative_code_type**, типа данных о дате **date_data_type**, типа данных о времени **time_data_type** или типа данных о времени и дате **date_time_data_type,** ограничение накладывается на (уникальную) строку, значения которой — это значение указанного типа данных. Для неколичественного кодового типа **non_quantitative_code_type** ограничение накладывается на код.

Для значений, принадлежащих переводимому строчному типу **translatable_string_type**, ограничение накладывается на строку, представленную на исходном языке, на котором определена область значений свойства. Исходный язык может быть определен атрибутом **source_language** административных данных **administrative_data** свойства. Если данный атрибут не существует, то данный исходный язык предполагается известным пользователю словаря.

Примечание 2 — Для неколичественного кодового типа **non_quantitative_code_type**, типа данных о дате **date_data_type**, типа данных о времени **time_data_type** или типа данных о времени и дате **date_time_data_type** шаблон должен соответствовать с неформальным требованиям, определенным соответствующими типами данных.

Пример представления на языке EXPRESS:

```
*)
ENTITY string_pattern_constraint
SUBTYPE OF (domain_constraint);
        pattern: STRING;
END_ENTITY; -- string_pattern_constraint
(*
```

Определения атрибутов:

pattern: шаблон значений строки, допустимых в качестве значений свойства, идентифицированного атрибутом **constrained_property**.

Дополнительное пояснение к тексту программы:

IP1: синтаксис шаблона **pattern** должен удовлетворять требованиям синтаксиса регулярных выражений языка XML и ассоциированных алгоритмов сравнения, определенных Схемой XML Часть 2: «Рекомендованные типы данных».

Пример — Шаблон Схемы XML «[0—9]{4}\-[0—9]{2}\-[0—9]{2}» соответствует выражению языка структурированных запросов SQL «[0—9][0—9][0—9][0—9]-[0—9][0—9]-[0—9][0—9]». Он допускает сравнение строк типа «2009-05-31».

F.5.2.15 Cardinality_constraint

Сущность **cardinality_constraint** ограничивает кардинальное число комплексного типа данных.

Примечание 1 — Результирующий диапазон значений кардинального числа — это пересечение уже существующих диапазонов значений кардинального числа и диапазона, определенного атрибутом **cardinality_constraint**.

Примечание 2 — Ограничения **cardinality_constraint** недопустимы для типа данных **array_type**.

Пример представления на языке EXPRESS:

```
*)
ENTITY cardinality_constraint
SUBTYPE OF (domain_constraint);
        bound_1: OPTIONAL INTEGER;
        bound_2: OPTIONAL INTEGER;
WHERE
        WR1: (bound_1 >= 0) AND (bound_2 >= bound_1);
END_ENTITY; -- cardinality_constraint
(*
```

Определения атрибутов:

bound_1: нижняя граница кардинального числа.

bound_2: верхняя граница кардинального числа.

Примечание 3 — Если граница **bound_1** не задана, то минимальное кардинальное число равно 0. Если граница **bound_2** не задана, то максимальное кардинальное число не ограничено.

Пояснения к тексту программы:

WR1: атрибуты **bound_1** и **bound_2** задают корректные границы.

F.5.3 Определения типа стандартной схемы ограничений класса ISO13584_IEC61360_class_constraint_schema

Данный подраздел определяет типы стандартной схемы ограничений класса **ISO13584_IEC61360_class_constraint_schema**.

F.5.3.1 Ограничение и идентификатор ограничения (Constraint_or_constraint_id)

Сущность **constraint_or_constraint_id** задает либо само ограничение **constraint,** либо идентификатор ограничения **constraint_identifier**.

Пример представления на языке EXPRESS:

```
*)
TYPE constraint_or_constraint_id =
        SELECT (constraint, constraint_identifier);
END_TYPE; -- constraint_or_constraint_id
(*
```

F.5.4 Определения функций стандартной схемы ограничений класса ISO13584_IEC61360_class_constraint_schema

Данный подраздел определяет функции стандартной схемы ограничений класса **ISO13584_IEC61360_class_constraint_schema.**

F.5.4.1 Функция определения целого значения в диапазоне (Integer_value_in_range)

Функция **integer_value_in_range** вычисляет целое значение, принадлежащее диапазону целых значений, определенному своей нижней границей и своей верхней границей. Если границы диапазона не определены, то значение функции также не определено.

Пример представления на языке EXPRESS:

```
*)
FUNCTION integer_values_in_range(
        low_bound, high_bound: INTEGER): SET OF INTEGER;
LOCAL
        i: INTEGER;
        result: SET OF INTEGER:= [ ];
END_LOCAL;
        IF EXISTS (low_bound) AND EXISTS (high_bound)
        THEN
                REPEAT i:= low_bound TO high_bound;
                        result:= result + [i];
                END_REPEAT;
                RETURN(result);
        ELSE
                RETURN(?);
        END_IF;
END_FUNCTION; -- integer_values_in_range
(*
```

F.5.4.2 Функция проверки предварительного условия (Correct_precondition)

Функция **correct_precondition** проверяет тот факт, что предварительное условие ограничения управления конфигурацией **configuration_control_constraint,** определенное атрибутом **cons,** использует только свойства, применимые в классе **cl**. Функция возвращает логическое значение. Это значение неизвестно, если полный набор применимых свойств в классе не может быть вычислен.

Пример представления на языке EXPRESS:

```
*)
FUNCTION correct_precondition(
        cons: configuration_control_constraint; cl:class): LOGICAL;
LOCAL
        prop: SET OF property_BSU:= [ ];
END_LOCAL;
        REPEAT i:= 1 to SIZEOF (cons.precondition);
                prop:= prop + cons.precondition[i].referenced_property;
        END_REPEAT;

        IF prop <= cl.known_applicable_properties
        THEN RETURN (TRUE);
        ELSE
                IF all_class_descriptions_reachable(cl.identified_by)
                THEN RETURN (FALSE);
                ELSE RETURN (UNKNOWN);
                END_IF;
        END_IF;
END_FUNCTION; -- correct_precondition
(*
```

F.5.4.3 Функция проверки корректности типа ограничения (Correct_constraint_type)

Функция **correct_constraint_type** проверяет тот факт, что ограничение на область **domain_constraint,** определенное атрибутом **cons,** совместимо с типом данных **data_type,** определенным атрибутом **typ.** Функция возвращает логическое значение. Это значение неизвестно, если тип **domain_constraint,** определенный атрибутом **cons,** не является одним из подтипов, определенных стандартной схемой ограничений класса **ISO13584_IEC61360_class_constraint_schema.**

Пример представления на языке EXPRESS:

```
*)
FUNCTION correct_constraint_type(
        cons: domain_constraint; typ:data_type): LOGICAL;
(*case subclass constraint*)

IF ('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
        +'SUBCLASS_CONSTRAINT') IN TYPEOF(cons)
THEN
        (*the data type shall be class_reference_type*)
        IF NOT
                ('ISO13584_IEC61360_DICTIONARY_SCHEMA.CLASS_REFERENCE_TYPE'
                IN TYPEOF (typ))
```

```
        THEN RETURN(FALSE);
        END_IF;

        (*the cons.subclasses shall consist of subclasses for the class
        that defined the initial domain of typ.*)
        IF NOT (QUERY (sc <* cons.subclasses |
                        definition_available_implies
                (sc,definition_available_implies
                (typ\class_reference_type.domain,is_subclass(sc.definition[1]
                , typ\class_reference_type.domain.definition[1])))= false)
                = [ ] )
        THEN RETURN(FALSE);
        END_IF;

        RETURN (TRUE);
END_IF;
(*case entity subtype constraint *)

IF (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
        +'ENTITY_SUBTYPE_CONSTRAINT') IN TYPEOF (CONS))
THEN

(* the data type is a class_reference_type*)
        IF NOT (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
        +'.ENTITY_INSTANCE_TYPE') IN TYPEOF (typ))
        THEN RETURN(FALSE);
        END_IF;
(* the subtype_name shall define a subtype for the entity_instance_type of the constrained        *)
        IF NOT (cons\entity_subtype_constraint.subtype_names
                >= typ\entity_instance_type.type_name)
        THEN RETURN(FALSE);
        END_IF;
        RETURN (TRUE);
        END_IF;

(*case enumeration_constraint*)
IF ('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
        +'.ENUMERATION_CONSTRAINT') IN TYPEOF (CONS)
THEN
(* all the values belonging to the subset of values shall be compatible with the
typ data type *)
        IF (QUERY (val<*cons.subset |
                NOT compatible_data_type_and_value (typ, val))<> [ ] )
        THEN RETURN(FALSE);
        END_IF;
RETURN (TRUE);
END_IF;

(*case range_constraint *)
IF ('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA.RANGE_CONSTRAINT'
        IN TYPEOF (CONS))
THEN

(*if the data type is an integer_type then min_value and max_value shall
be INTEGERs.*)
        IF ('ISO13584_IEC61360_DICTIONARY_SCHEMA.INTEGER_TYPE'
                IN TYPEOF (typ)) AND
                NOT ('INTEGER' IN TYPEOF (cons.min_value))
        THEN RETURN(FALSE);
        END_IF;

(*if the data type is a rational_type then min_value and max_value shall be rational.*)
        IF ('ISO13584_IEC61360_DICTIONARY_SCHEMA.RATIONAL_TYPE'
```

```
            IN TYPEOF (typ)) AND
NOT ('ISO13584_INSTANCE_RESOURCE_SCHEMA.RATIONAL_VALUE' IN TYPEOF
(cons.min_value))
THEN RETURN(FALSE);
END_IF;

(*if the data type is a real_type then min_value and max_value shall be REALs. *)
      IF ('ISO13584_IEC61360_DICTIONARY_SCHEMA.REAL_TYPE'
            IN TYPEOF (typ)) AND NOT ('REAL' IN TYPEOF (cons.min_value))
      THEN RETURN(FALSE);
      END_IF;

(*all values of the range shall belong to the allowed values defined by the type. *)
      IF (('ISO13584_IEC61360_DICTIONARY_SCHEMA'
            + '.NON_QUANTITATIVE_INT_TYPE') IN TYPEOF (typ))
      AND NOT
            (integer_values_in_range(cons.min_value, cons.max_value)
            <= allowed_values_integer_types (typ))
      THEN RETURN(FALSE);
      END_IF;

RETURN (TRUE);
END_IF;
(*case entity string_size_constraint *)

IF ('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
      +'.STRING_SIZE_CONSTRAINT') IN TYPEOF (CONS)
THEN

(* the data type shall be a string_type or any of its subtypes      *)
      IF NOT ('ISO13584_IEC61360_DICTIONARY_SCHEMA.STRING_TYPE'
            IN TYPEOF (typ))
      THEN RETURN(FALSE);
      END_IF;

RETURN (TRUE);
END_IF;

(*case entity string_pattern_constraint *)

IF ('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
      +'.STRING_PATTERN_CONSTRAINT') IN TYPEOF (CONS)
THEN

(* the data type shall be a string_type or any of its subtypes *)
      IF NOT ('ISO13584_IEC61360_DICTIONARY_SCHEMA.STRING_TYPE'
            IN TYPEOF (typ))
      THEN RETURN(FALSE);
      END_IF;
RETURN (TRUE);
END_IF;

(*case entity cardinality_constraint *)
IF ('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
      +'.CARDINALITY_CONSTRAINT') IN TYPEOF (CONS)
THEN
      (* the data type shall be an aggregate type but not an array *)
      IF (NOT(
      ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA'
      + '.ENTITY_INSTANCE_TYPE_FOR_AGGREGATE')
      IN TYPEOF(typ)))
THEN
      RETURN(FALSE);
```

```
END_IF;
        IF ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA'
        + '.ARRAY_TYPE' IN TYPEOF(typ.type_structure))
        THEN
                RETURN(FALSE);
        END_IF;
        RETURN (TRUE);
END_IF;

RETURN (UNKNOWN);

END_FUNCTION; -- correct_constraint_type
(*
```

F.5.4.4 Функция проверки совместимости типа данных и значения (Compatible_**data_type**_and_value)

Функция **compatible_data_type_and_value** проверяет тот факт, что тип значения **val** атрибута **primitive_value** совместим с типом, определенным атрибутом **dom.** Функция возвращает логическое значение, равное TRUE, если типы совместимы. Это значение равно FALSE, если типы несовместимы. Данная функция возвращает значение UNKNOWN, когда тип данных **val** соответствует типу неуправляемой реализации **uncontrolled_instance_value**(см. ИСО 13584-24:2003), или если рассматриваемый тип не относится к типам, определенным стандартной схемой ресурса реализаций **ISO13584_instance_resource_schema.**

Примечание — Значение переменной **val** может существовать и может не существовать.

Пример представления на языке EXPRESS:

```
*)
FUNCTION compatible_data_type_and_value(dom: data_type;
        val: primitive_value): LOGICAL;

LOCAL
        temp: class_BSU;
        set_string: SET OF STRING:= [ ];
        set_integer: SET OF INTEGER:= [ ];
        code_type: non_quantitative_code_type;
        int_type: non_quantitative_int_type;
END_LOCAL;

(* The following express statements deal with simple types *)
IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.INTEGER_VALUE' IN TYPEOF(val))
THEN
        IF ('ISO13584_IEC61360_DICTIONARY_SCHEMA.' +
                'NON_QUANTITATIVE_INT_TYPE' IN TYPEOF (dom))
        THEN
                set_integer:= [ ];
                int_type:= dom;
                REPEAT j:= 1 TO SIZEOF(int_type.domain.its_values);
                        set_integer:= set_integer +
                                int_type.domain.its_values[j].value_code;
                END_REPEAT;
                RETURN(val IN set_integer);
        ELSE
                RETURN(('ISO13584_IEC61360_DICTIONARY_SCHEMA.INT_TYPE'
                        IN TYPEOF (dom)) OR
                        (('ISO13584_IEC61360_DICTIONARY_SCHEMA.NUMBER_TYPE'
                        IN TYPEOF (dom))
                        AND NOT(('ISO13584_IEC61360_DICTIONARY_SCHEMA.REAL_TYPE'
                        IN TYPEOF (dom))
                        OR ('ISO13584_IEC61360_DICTIONARY_SCHEMA.RATIONAL_TYPE'
                        IN TYPEOF (dom)))));
        END_IF;
END_IF;
IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.REAL_VALUE' IN TYPEOF(val))
THEN
        RETURN(('ISO13584_IEC61360_DICTIONARY_SCHEMA.REAL_TYPE'
                IN TYPEOF (dom)) OR
```

```
                ((‘ISO13584_IEC61360_DICTIONARY_SCHEMA.NUMBER_TYPE’
                IN TYPEOF (dom))
                AND NOT((‘ISO13584_IEC61360_DICTIONARY_SCHEMA.INT_TYPE’
                IN TYPEOF (dom))
                OR (‘ISO13584_IEC61360_DICTIONARY_SCHEMA.RATIONAL_TYPE’
                        IN TYPEOF (dom))));
        END_IF;

        IF (‘ISO13584_INSTANCE_RESOURCE_SCHEMA.RATIONAL_VALUE’ IN TYPEOF(val))
        THEN
                RETURN((‘ISO13584_IEC61360_DICTIONARY_SCHEMA.RATIONAL _TYPE’
                        IN TYPEOF (dom)) OR
                        ((‘ISO13584_IEC61360_DICTIONARY_SCHEMA.NUMBER_TYPE’
                        IN TYPEOF (dom))
                        AND NOT((‘ISO13584_IEC61360_DICTIONARY_SCHEMA.INT_TYPE’
                        IN TYPEOF (dom))
                        OR (‘ISO13584_IEC61360_DICTIONARY_SCHEMA.REAL_TYPE’
                                IN TYPEOF (dom))));
        END_IF;

        IF (‘ISO13584_INSTANCE_RESOURCE_SCHEMA.String_value’
                IN TYPEOF(val))
        THEN
                IF ((‘ISO13584_IEC61360_DICTIONARY_SCHEMA’ +
‘.                      NON_QUANTITATIVE_CODE_TYPE’) IN TYPEOF (dom))
                THEN
                        set_string:= [ ];
                        code_type:= dom;
                        REPEAT j:= 1 TO SIZEOF(code_type.domain.its_values);
                                set_string:= set_string +
                                        code_type.domain.its_values[j].value_code;
                        END_REPEAT;
                        RETURN(val IN set_string);
                ELSE
                        RETURN(‘ISO13584_IEC61360_DICTIONARY_SCHEMA’ +
                                ‘.STRING_TYPE’ IN TYPEOF (dom));
                END_IF;
        END_IF;
        IF (‘ISO13584_INSTANCE_RESOURCE_SCHEMA.Translated_string_value’
                IN TYPEOF(val))
        THEN
                RETURN(‘ISO13584_IEC61360_DICTIONARY_SCHEMA’ +
                        ‘.Translatable_string_type’ IN TYPEOF (dom));
        END_IF;

        (* The following express statements deal with complex types *)

        IF ‘ISO13584_INSTANCE_RESOURCE_SCHEMA.DIC_CLASS_INSTANCE’
                IN TYPEOF(val)
        THEN
                IF (‘ISO13584_IEC61360_DICTIONARY_SCHEMA.CLASS_REFERENCE_TYPE’
                        IN TYPEOF (dom))
                THEN
                        temp:= dom.domain;
                        RETURN(compatible_class_and_class(temp,
                                val\dic_class_instance.class_def));
                ELSE
                        RETURN(FALSE);
                END_IF;
        END_IF;

        IF ‘ISO13584_INSTANCE_RESOURCE_SCHEMA.LEVEL_SPEC_VALUE’ IN TYPEOF(val)THEN
                IF (‘ISO13584_IEC61360_DICTIONARY_SCHEMA.LEVEL_TYPE’
```

```
            IN TYPEOF (dom))
        THEN
            RETURN(compatible_level_type_and_instance(
                dom.levels,
                TYPEOF(dom.value_type),
                val));
        ELSE
            RETURN(FALSE);
        END_IF;
    END_IF;

    (* The following express statements deal with aggregate types  *)

    IF 'ISO13584_AGGREGATE_VALUE_SCHEMA.AGGREGATE_ENTITY_INSTANCE_VALUE' IN
    TYPEOF(val) THEN
        IF (NOT(
        'ISO13584_IEC61360_DICTIONARY_SCHEMA.ENTITY_INSTANCE_TYPE'
            IN TYPEOF(dom)))
        THEN
            RETURN(FALSE);
        END_IF;

        IF (NOT(
            'ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA'
            + '.AGGREGATE_TYPE' IN dom.type_name))
        THEN
            RETURN(FALSE);
        END_IF;

        RETURN(compatible_aggregate_type_and_value(dom, val));
    END_IF;

    IF 'ISO13584_INSTANCE_RESOURCE_SCHEMA.ENTITY_INSTANCE_VALUE'
        IN TYPEOF(val)
    THEN
        IF 'ISO13584_INSTANCE_RESOURCE_SCHEMA' +
            '.UNCONTROLLED_ENTITY_INSTANCE_VALUE'
            IN TYPEOF(val)
        THEN
            RETURN(UNKNOWN);
        END_IF;
        IF ('ISO13584_IEC61360_DICTIONARY_SCHEMA.ENTITY_INSTANCE_TYPE'
            IN TYPEOF (dom))
            AND (dom.type_name <= TYPEOF(val))
        THEN
            RETURN(TRUE);
        ELSE
            RETURN(FALSE);
        END_IF;
    END_IF;

    RETURN(UNKNOWN);

    END_FUNCTION; -- compatible_data_type_and_value
    (*
```

F.5.5 Определение правил стандартной схемы ограничений класса ISO13584_IEC61360_class_constraint_schema

Данный подраздел определяет правила стандартной схемы ограничений класса **ISO13584_IEC61360_class_constraint_schema.**

F.5.5.1 Уникальный идентификатор ограничения (Unique_constraint_id)

Правило **unique_constraint_id** подтверждает, что два идентификатора ограничения **constraint_identifier,** ассоциированные с двумя различными ограничениями, имеют различные значения.

Пример представления на языке EXPRESS:

```
*)
RULE unique_constraint_id FOR(constraint);
WHERE
    QUERY(c1 <* constraint |
        SIZEOF(QUERY(c2 <* constraint |
        c1.constraint_id = c2.constraint_id))>1) = [ ];
END_RULE; — unique_constraint_id
(*
*)
END_SCHEMA; — ISO13584_IEC61360_class_constraint_schema

(*
```

F.6 Стандартная условная схема класса предметов ISO13584_IEC61360_item_class_case_of_schema

Данный раздел определяет требования к условной схеме класса предметов **ISO13584_IEC61360_item_class_case_of_schema**. Нижеследующая декларация языка EXPRESS представляет блок схемы **ISO13584_IEC61360_item_class_case_of_schema** и задает необходимые внешние ссылки.

Пример представления на языке EXPRESS:

```
*)
SCHEMA ISO13584_IEC61360_item_class_case_of_schema;

REFERENCE FROM ISO13584_IEC61360_dictionary_schema
    (all_class_descriptions_reachable,
    class,
    class_BSU,
    data_type_BSU,
    item_class,
    property_BSU);
REFERENCE FROM ISO13584_IEC61360_class_constraint_schema
    (constraint,
    integrity_constraint,
    context_restriction_constraint,
    property_constraint,
    domain_constraint);
REFERENCE FROM ISO13584_extended_dictionary_schema
    (document_BSU,
    table_BSU,
    visible_properties,
    applicable_properties,
    visible_types,
    applicable_types,
    data_type_named_type);
(*
```

Примечание — Схематики, на которые производятся ссылки выше, можно найти в нижеследующих документах:

ISO13584_IEC61360_dictionary_schema МЭК 61360-2
(схема дублирована для удобства в данном документе).
ISO13584_IEC61360_class_constraint_schema МЭК 61360-2
(схема дублирована для удобства в данном документе).
ISO13584_extended_dictionary_schema ИСО 13584-24:2003

F.6.1 Введение в стандартную условную схему класса предметов ISO13584_IEC61360_item_class_case_of_schema

Полная общая словарная модель ИСО/МЭК поделена между несколькими документами по принципу блочности. Модель ядра онтологии продукта определена в МЭК 61360-2 и дублирована в настоящем стандарте. Ресурс расширения данной модели, включая представление реализации, представление документа, функциональную модель, функциональные виды и представление таблицы, определен в ИСО 13584-24:2003. Различные стандартные уровни практической реализации полной стандартной модели ИСО/МЭК, называемые классами соответствия, определены в ИСО 13584-25 и дублируются (для справочных целей) в МЭК 61360-5. Первый уровень точно соответствует содержанию настоящего стандарта, главным образом, в части ресурса для агрегатно-структурированных значений, определенных в ИСО 13584-25. Прочие классы соответствия включают, главным образом, ресурсы из ИСО 13584-24:2003.

Определение класса предметов, как другого условного класса предметов, все чаще используется в приложениях, основанных на общей словарной модели ИСО 13584/МЭК 61360. Более того, данная версия настоящего стандарта требует изменения информационной модели данного понятия. Поэтому принято решение перенести соответствующую сущность языка EXPRESS, называемую условный класс предметов **item_class_case_of,** и ее суперкласс, называемый априорным семантическим соотношением **a_priori_semantic_relationship,** из ИСО 13584-24:2003 в настоящий стандарт. Указанные сущности включены в новую схему, называемую стандартной условной схемой класса предметов **ISO13584_IEC61360_item_class_case_of_schema.** ИСО 13584-24:2003 и ИСО 13584-25 обновляются путем внесения технических поправок в установленном порядке.

F.6.2 Определения сущностей стандартной условной схемы класса предметов ISO13584_IEC61360_ item_class_case_of_schema

F.6.2.1 Априорное семантическое соотношение

Априорное семантическое соотношение **a_priori_semantic_relationship** — это абстрактный класс, определенный на базе других классов. Данный класс может импортировать свойства, типы данных, таблицы и документы, содержащиеся в указанных классах. Он также импортирует все ограничения, накладывающие условия на область импортированных свойств в классе, из которого они импортированы. Данный абстрактный ресурс разбивается классами на подтипы. Если класс специализирует некоторое априорное семантическое соотношение **a_priori_ semantic_relationship,** то свойства, типы данных, таблицы или документы, определения которых импортированы по наследству с помощью **a_priori_semantic_relationship,** становятся применимыми в классе, который их импортирует. В частности, свойства и типы данных, импортированные таким образом, могут использоваться для описаний реализаций класса. Факт наличия аспекта (продукта), соответствующего каждому импортированному свойству, является необходимым критерием членства в классе.

Примечание 1 — Все импортированные свойства и типы данных становятся непосредственно применимыми без обеспечения их видимости. Таким образом, они не возвращаются функцией вычисления известных видимых свойств **compute_known_visible_properties** или функцией вычисления известных видимых типов данных **compute_known_visible_data_type**.

Примечание 2 — Соотношение наследственности — это хорошо известный пример установления семантического соотношения между классами, моделируемыми в соответствии с имеющейся объектно-ориентированной парадигмой. Все свойства и прочие ресурсы, определеные в классе, обычно задаются неявно во всех своих подклассах. Данное соотношение используется в Серии стандартов ИСО 13584, где все свойства, типы данных, таблицы или документы, видимые (и, соответственно, применимые) в некоторых классах, являются неявно видимыми (и, соответственно, применимыми) во всех своих подклассах. Как обычно, в ИСО 13584 данное наследование является неявным (т. е. не объявлено с помощью априорного семантического соотношения **a_priori_semantic_ relationship**) и глобальным (т. е. все свойства и типы данных унаследованы всеми его подклассами). Априорное семантическое соотношение **a_priori_semantic_relationship** дает возможность определять прочие семантические соотношения, используемые в области приложений ИСО 13584, и, в частности, условные соотношения, допускающие явное и частичное импортирование свойств и прочих ресурсов, определенных в классе.

Пример представления на языке EXPRESS:

```
*)
ENTITY a_priori_semantic_relationship
ABSTRACT SUPERTYPE
SUBTYPE OF(class);
        referenced_classes: SET [1:?] OF class_BSU;
        referenced_properties: LIST [0:?] OF property_BSU;
        referenced_data_types: SET [0:?] OF data_type_BSU;
        referenced_tables: SET [0:?] OF table_BSU;
        referenced_documents: SET [0:?] OF document_BSU;
        referenced_constraints: SET [0:?] OF constraint_or_constraint_id;
WHERE
        WR1: QUERY (cons <* SELF.referenced_constraints
                | NOT(('ISO13584_IEC61360_DICTIONARY_SCHEMA' +
                '.ISO_29002_IRDI_type') IN TYPEOF(cons))
                AND NOT (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
                +'.PROPERTY_CONSTRAINT') IN TYPEOF (cons)))
                = [ ];
        WR2: QUERY (cons <* SELF.referenced_constraints
                | (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
                +'.PROPERTY_CONSTRAINT') IN TYPEOF (cons))
                AND NOT (cons\property_constraint.constrained_property
                IN SELF.referenced_properties))
                = [ ];
        WR3: compute_known_referenced_property_constraints(SELF)
                <= SELF.referenced_constraints;
```

```
        WR4: QUERY(prop <* SELF.referenced_properties
             | QUERY(cl <* SELF.referenced_classes
             | visible_properties(cl, [prop])
             OR applicable_properties(cl, [prop]))
             = [ ] ) = [ ];
        WR5: QUERY(typ <* SELF.referenced_data_types
             | QUERY(cl <* SELF.referenced_classes
             | visible_types(cl, [typ])
             OR applicable_types(cl, [typ]))
             = [ ] ) = [ ];
    END_ENTITY; -- a_priori_semantic_relationship
    (*
```

Определения атрибутов:

referenced_class: классы, откуда импортируются свойства, типы данных, таблицы и документы.

Примечание 3 — Класс, из которого импортируются свойства, типы данных, таблицы и документы, не может быть выведен по идентификации импортированных свойств, типов данных, таблиц и документов, потому что они сами могут быть импортированы из класса, где они унаследованы или импортированы. Например, по МЭК 61360-DB «входное напряжение» — это свойство, видимое на корневом уровне классификации Международной Электротехнической Комиссии (МЭК). Если поставщик класса импортирует свойство «входное напряжение» из класса «транзисторы» МЭК, то: (1) поставщик класса определяет транзистор, (2) указанные транзисторы описаны с помощью свойства «входное напряжение».

referenced_properties: свойства, определения которых импортированы с помощью сущности **a_priori_semantic_relationship**.

Примечание 4 — Списочный порядок определяет порядок по умолчанию для представления импортированных свойств во время процесса обеспечения доступа пользователя к различным подтипам априорного семантического соотношения **a_priori_semantic_relationship**.

referenced_data_type: типы данных, определения которых импортированы с помощью сущности **a_priori_semantic_relationship**.

referenced_tables: таблицы, определения которых импортированы с помощью сущности **a_priori_semantic_relationship**.

Примечание 5 — Подробные описания ресурсов и правил пользования таблицами даны в ИСО 13584-24:2003. Они не используются ни в настоящем стандарте, ни в интегрированных моделях, задокументированных в ИСО 13584-32 (OntoML) и ИСО 13584-25.

referenced_document: документы, определения которых импортированы с помощью сущности **a_priori_semantic_relationship**.

Примечание 6 — Детальные описания ресурсов и правил пользования документами определены в ИСО 13584-24:2003. Они используются в интегрированных моделях, задокументированных в ИСО 13584-32 (OntoML) и ИСО 13584-25.

referenced_constraint: ссылочные ограничения свойств **property_constraints,** накладываемые на различные импортированные свойства.

Примечание 7 — В отличие от других ссылочных сущностей ссылочные ограничения **referenced_constraints** не могут быть выбраны при разработке априорного семантического соотношения **a_priori_semantic_relationship**. Указанные ограничения — это все ограничения для всех свойств, определенных атрибутом **referenced_properties** в любом классе атрибута **referenced_class**.

Пояснения к тексту программы:

WR1: все ссылочные ограничения **referenced_constraints,** не относящиеся к МИРД, должны быть ограничениями свойств **property_constraints**.

WR2: все ссылочные ограничения **referenced_constraints** должны ограничивать свойства, импортированные с помощью атрибута **referenced_properties**.

WR3: все ограничения свойств **property_constraints**, ограничивающие одно из ссылочных свойств **referenced_properties** в любом ссылочном классе **referenced_classes**, должны быть импортированы с помощью атрибута **referenced_constraints.**

WR4: импортированные свойства, определенные атрибутом **referenced_properties,** должны быть видимыми или применимыми в одном из классов, принадлежащих атрибуту **referenced_class**.

WR5: импортированные типы, определенные атрибутом **referenced_data_types,** должны быть видимыми или применимыми в одном из классов, принадлежащих атрибуту **referenced_classes**.

Дополнительные пояснения:

IP1: все ограничения, представленные идентификаторами **constraint_identifiers** в наборе **referenced_constraints,** должны соответствовать требованиям ограничений свойств **property_constraints**, наложенных на одно из ссылочных свойств **referenced_properties** в одном из рассматриваемых ссылочных классов **referenced_classes**. Такие ограничения не могут быть представлены (в том же наборе ссылочных ограничений **referenced_constraint)** одновременно и как ограничение свойства **property_constraint,** и как идентификатор ограничения **constraint_identifier.**

Примечание 8 — Ограничение, представленное как ограничение свойства **property_constraint** в одном из ссылочных классов **referenced_classes**, может быть представлено в наборе ссылочных ограничений **referenced_constraints** либо как ограничение свойства **property_constraint**, либо как идентификатор ограничения **constraint_identifier**.

IP2: все ограничения, представленные идентификатором ограничения **constraint_identifier** в одном из рассматриваемых ссылочных классов **referenced_classes** (с соответствующим ограничением свойства **property_constraint,** наложенным на одно из свойств, импортированных с помощью атрибута **referenced_properties),** должны быть представлены идентификатором ограничения **constraint_identifier** в наборе ссылочных ограничений **referenced_constraints**.

Примечание 9 — Эти два неформальных правила гарантируют, что рассматриваемый набор ссылочных ограничений **referenced_constraints** является объединением наборов ограничений свойств **property_constraint**, определенных в различных ссылочных классах **referenced_classes** с ограниченными свойствами **constrained_property**, принадлежащими набору ссылочных свойств **referenced_properties**, даже если контекст обмена не содержит определений всех классов, привлеченных к априорному семантическому соотношению **a_priori_semantic_relationship**, и если некоторые ограничения представлены только их идентификаторами **constraint_identifier**.

F.6.2.2 Условный класс предметов (Item_class_case_of)

Сущность **item_class_case_of** дает описание класса предметов, определенного как условную комбинацию некоторых других классов предметов.

Примечание 1 — Сущность **item_class_case_of** определяет априорное семантическое соотношение.

Пример представления на языке EXPRESS:

```
*)
ENTITY item_class_case_of
SUBTYPE OF(item_class, a_priori_semantic_relationship);
        is_case_of: SET [1:?] OF class_BSU;
        imported_properties: LIST [0:?] OF property_BSU;
        imported_types: SET [0:?]OF data_type_BSU;
        imported_tables: SET [0:?] OF table_BSU;
        imported_documents: SET [0:?] OF document_BSU;
        imported_constraints: SET [0:?] OF constraint_or_constraint_id;
DERIVE
        SELF\a_priori_semantic_relationship.referenced_classes:
                SET [1:?] OF class_BSU:= SELF.is_case_of;
        SELF\a_priori_semantic_relationship.referenced_properties:
                LIST [0:?] OF property_BSU:= SELF.imported_properties;
        SELF\a_priori_semantic_relationship.referenced_data_types:
                SET [0:?] OF data_type_BSU:= SELF.imported_types;
        SELF\a_priori_semantic_relationship.referenced_tables:
                SET [0:?] OF table_BSU:= SELF.imported_tables;
        SELF\a_priori_semantic_relationship.referenced_documents:
                SET [0:?] OF document_BSU:= SELF.imported_documents;
        SELF\a_priori_semantic_relationship.referenced_constraints:
                SET [0:?] OF property_constraint
        := SELF.imported_constraints;
WHERE
        WR1: superclass_of_item_is_item(SELF);
        WR2: check_is_case_of_referenced_classes_definition(SELF);
        WR3: QUERY(p <* SELF\class.sub_class_properties
                | NOT((p IN SELF.described_by)
                OR (p IN SELF.imported_properties))) = [ ];
        WR4: QUERY(p <* SELF\class.sub_class_properties
                | (p IN SELF.imported_properties)
                AND (QUERY(cl<*SELF.is_case_of
                | all_class_descriptions_reachable(cl) AND
```

```
        (p IN compute_known_applicable_properties(cl)) AND
        (NOT is_class_valued_property(p, cl)))<>[ ] ))
        =[ ];
    WR5: QUERY(ccv <* SELF\class.class_constant_values
        I (ccv.super_class_defined_property
        IN SELF.imported_properties)
        AND (QUERY(cl<*SELF.is_case_of
        I all_class_descriptions_reachable(cl) AND
        (ccv.super_class_defined_property
        IN compute_known_applicable_properties(cl)) AND
        (QUERY (v<*class_value_assigned(
        ccv.super_class_defined_property, cl)
        I v<> ccv.assigned_value) <> [ ] ))<>[ ] ))
        =[ ];
    WR6: QUERY(prop <* imported_properties
        I (QUERY(cl<*SELF.is_case_of
        I is_class_valued_property(prop, cl)) <>[ ] )
        AND NOT is_class_valued_property(prop, SELF.identified_by))
        =[ ];
    WR7: QUERY(ccv <* SELF\class.class_constant_values
        I QUERY(cl<*SELF.is_case_of
        I (class_value_assigned
        (ccv.super_class_defined_property, cl) <> [ ] )
        AND (QUERY(v <* class_value_assigned
        (ccv.super_class_defined_property, cl)
        I v <> ccv. assigned_value)<>[ ] )) <> [ ] )
        =[ ];
END_ENTITY; -- item_class_case_of
(*
```

Определения атрибутов:

is_case_of: классы предметов **item_class(es),** из которых комбинируется рассматриваемый класс предметов **item_class**.

imported_properties: список свойств, импортированных из классов предметов **item_class(es),** из которых комбинируется рассматриваемый класс предметов **item_class**.

imported_type: набор тип данных, импортированных из классов предметов **item_class(es)**, из которых комбинируется рассматриваемый класс предметов **item_class.**

imported_tables: набор базовых семантических единиц таблиц **table_BSUs,** импортированных из классов предметов **item_class(es),** из которых комбинируется рассматриваемый **item_class**.

imported_document: набор базовых семантических единиц документов **document_BSUs,** импортированных из классов предметов **item_class(es),** из которых комбинируется рассматриваемый **item_class**.

imported_constraint: набор ограничений свойств **property_constraint** или идентификаторов ограничений **constraint_id,** импортированных из классов предметов **item_class(es),** из которых комбинируется рассматриваемый **item_class**.

Примечание 2 — В отличие от других импортированных сущностей, импортированные ограничения **imported_constraints** не могут быть выбраны при разработке условного класса предметов **item_class_case_of**. Указанные ограничения — это все ограничения, наложенные на область значений любого из свойств, определенного импортированными свойствами **imported_properties** в классах атрибута **is_case_of**, из которых они импортированы. Это определено областью применения априорного семантического соотношения **a_priori_semantic_relationship**.

Пояснения к тексту программы:

WR1: суперклассом для условного класса предметов **item_class_case_of** должен быть класс предметов **item_class.**

WR2: условный класс предметов **item_class_case_of** должен быть комбинацией классов предметов **item_class(es).**

WR3: свойства подкласса **sub_class_properties** должны принадлежать либо списку описаний **described_by**, либо списку импортированных свойств **imported_properties**.

WR4: все свойства со значением класса, объявленные с помощью свойств подкласса **sub_class_properties (**являющихся импортированными свойствами **imported_properties),** должны быть свойствами со значением класса во всех комбинируемых классах, где они применимы.

WR5: значения, заданные импортированному свойству с помощью атрибута значения константы класса **class_constant_value** не должны различаться более, чем на возможное значение, заданное для того же самого свойства в рассматриваемых ссылочных классах.

WR6: все импортированные свойства **imported_properties,** являющиеся свойствами со значением класса, являющегося условной комбинацией других классов, должны быть свойствами со значением класса в рассматриваемом классе.

WR7: всем импортированным свойствам **imported_properties** со значением константы класса **class_constant_value** из условной комбинации других классов, должно быть задано одно и то же значение константы класса **class_constant_value** в рассматриваемом классе.

F.6.3 Определения функций стандартной схемы условного класса предметов ISO13584_IEC61360_item_class_case_of_schema

Данный подраздел содержит описания функций, на которые производится ссылка в разделах по месту для подтверждения непротиворечивости данных или для создания ресурсов приложений.

F.6.3.1 Функция вычисления ограничения известного свойства (Compute_known_property_constraints)

Функция **compute_known_property_constraints** вычисляет набор ограничений свойств **property_constraints**, применимых для свойств набора классов. Ограничения, представленные их идентификаторами, не вычисляются. Если определение класса недоступно, то функция возвращает только те ограничения свойств **property_constraints**, которые могут быть вычислены.

Примечание — Если словарное определение **dictionary_definition** класса недоступно в рассматриваемом контексте обмена (а контекст обмена библиотеки PLIB никогда не предполагается полным), то собственный суперкласс может быть неизвестен. Следовательно, ограничения, определенные данным суперклассом, не могут быть вычислены функцией **compute_known_property_constraint**. И наоборот, если все представительные суперклассы рассматриваемого класса доступны в том же контексте обмена, то все ограничения, применимые в данном классе, могут быть вычислены за один обход представительного дерева наследственности, даже если некоторые из указанных суперклассов импортируют свойства с помощью априорного семантического соотношения **a_priori_semantic_relationship,** такого как условный класс предметов **item_class_case_of.**

Пример представления на языке EXPRESS:

```
*)
FUNCTION compute_known_property_constraints(classes: SET OF class_BSU):
      SET OF property_constraint;
LOCAL
      s: SET OF property_constraint:= [ ];
END_LOCAL;

REPEAT nb:= 1 TO SIZEOF (classes);
      IF SIZEOF(classes[nb].definition)=1
      THEN
            REPEAT i:= 1 TO
                  SIZEOF(classes[nb].definition[1]\class.constraints);
                  IF (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
                  +'.PROPERTY_CONSTRAINT')

                  IN TYPEOF
                        (classes[nb].definition[1]\class.constraints[i]))
                  THEN
                  s:= s + classes[nb].definition[1]\class.constraints[i];
                  END_IF;
            END_REPEAT;

            IF (('ISO13584_IEC61360_ITEM_CLASS_CASE_OF_SCHEMA.'
                  + 'A_PRIORI_SEMANTIC_RELATIONSHIP')
                  IN TYPEOF (classes[nb].definition[1]))
            THEN
                  REPEAT i:= 1 TO
                  SIZEOF(classes[nb].definition[1]
                  \a_priori_semantic_relationship
                  .referenced_constraints);
                        IF (('ISO13584_IEC61360_CLASS_CONSTRAINT_SCHEMA'
                              +'.PROPERTY_CONSTRAINT') IN TYPEOF
                              (classes[nb].definition[1]
                              \a_priori_semantic_relationship
                              .referenced_constraints[i]))
                        THEN
                              s:= s + classes[nb].definition[1]
                              \a_priori_semantic_relationship
```

```
                                .referenced_constraints [i];
                        END_IF;
                    END_REPEAT;
                END_IF;

                IF EXISTS(classes[nb].definition[1]\class.its_superclass)
                THEN
                    s:= s + compute_known_property_constraints(
                        [classes[nb].definition[1]\class.its_superclass]);

                END_IF;

            END_IF;
        END_REPEAT;
        RETURN(s);

        END_FUNCTION; -- compute_known_property_constraints
        (*
```

F.6.3.2 Функция вычисления известного ссылочного ограничения свойства (Compute_known_referenced_property_constraint)

Функция **compute_known_referenced_property_constraint** вычисляет все ограничения свойства **property_constraint,** импортируемые априорным семантическим соотношением (**ap) a_priori_semantic_relationship,** путем вычисления всех ограничений, применимых к свойству, импортированному с помощью атрибута ссылочных свойств **referenced_properties (**из **ap),** определенному или унаследованному в любом ссылочном классе (**ap) referenced_class,** словарное определение **dictionary_definition** которого доступно в рассматриваемом контексте обмена.

Примечание 1 — В априорном семантическом соотношении **a_priori_semantic_relationship** все ограничения свойств **property_constraints**, определенные (унаследованные) классами, и на которые производится ссылка атрибутом **referenced_class**, применяемом к свойству, импортированному с помощью атрибута **referenced_properties** априорного семантического соотношения **a_priori_semantic_relationship**, должны быть импортированы с помощью своего атрибута ссылочного ограничения **referenced_constraints**.

Примечание 2 — Если словарное определение **dictionary_definition** класса, принадлежащее атрибуту **referenced_class** сущности **ap** не является доступным в рассматриваемом контексте обмена как сущность **ap** (а контекст обмена библиотеки PLIB никогда не предполагается полным), то ограничения, принадлежащие данному классу, не могут быть вычислены. Таким образом, результат функции **compute_known_referenced_property_constraint** может быть только подмножеством ограничений, импортируемых сущностью **ap**.

Примечание 3 — Если словарные определения **dictionary_definition** всех ссылочных классов **referenced_classes** сущности **ap** доступны в рассматриваемом контексте обмена, и если никакие ограничения не представлены единственным идентификатором ограничений **constraint_identifier**, то функция **compute_known_referenced_property_constraint** возвращает точно все ограничения, импортируемые сущностью **ap**.

Пример представления на языке EXPRESS:

```
        *)
        FUNCTION compute_known_referenced_property_constraints(
                    ap: a_priori_semantic_relationship):
                    SET OF property_constraint;
        LOCAL
            s: SET OF property_constraint:= [ ]; — result
            prop: SET OF property_BSU:=
                    list_to_set(ap.referenced_properties); — imported properties
            cl: SET OF class_BSU:=
                    ap.referenced_classes; — source of importation
            cons: SET OF property_constraint
                    := compute_known_property_constraints(cl);
                    - all those property_constraints existing in the various
                    - classes from cl that may be computed in the current
                    - exchange context.
        END_LOCAL;

            REPEAT n_cons:= 1 TO SIZEOF(cons);
                IF cons[n_cons].constrained_property IN prop
                THEN
                    s:= s + cons[n_cons];
```

```
            END_IF;
        END_REPEAT;
    RETURN(s);

    END_FUNCTION; -- compute_known_referenced_property_constraints
    (*
```

F.6.3.3 Функция проверки суперкласса предметов (Superclass_of_item_is_item)

Функция **superclass_of_item_is_item** проверяет тот факт, что суперкласс класса предметов (сущность **cl) item_class,** если он существует, также является классом предметов **item_class.**

Если класс, ассоциированный с базовой семантической единицей класса **class_BSU,** не может быть вычислен, то функция возвращает значение UNKNOWN.

Пример представления на языке EXPRESS:

```
    *)
    FUNCTION superclass_of_item_is_item(cl: item_class): LOGICAL;

    IF NOT EXISTS(cl\class.its_superclass)
    THEN
        RETURN(TRUE);
    END_IF;

    IF SIZEOF(cl\class.its_superclass.definition) = 0
    THEN
        RETURN(UNKNOWN);
    END_IF;

    RETURN(('ISO13584_IEC61360_DICTIONARY_SCHEMA.ITEM_CLASS')
        IN TYPEOF(cl\class.its_superclass.definition[1]));

    END_FUNCTION; -- superclass_of_item_is_item
    (*
```

F.6.3.4 Функция проверки определения условного ссылочного класса (Check_is_case_of_referenced_class_definition)

Функция **check_is_case_of_referenced_class_definition** возвращает значение TRUE, если тип условного класса предметов **item_class_case_of** из набора словарных определений ссылочных классов **is_case_of** совместим с рассматриваемой реализацией условного класса предметов (**cl**) **item_class_case_of**. В противном случае функция возвращает значение FALSE.

Пример представления на языке EXPRESS:

```
    *)
    FUNCTION check_is_case_of_referenced_classes_definition(
        cl: item_class_case_of): BOOLEAN;
    LOCAL
        class_def_ok: BOOLEAN:= TRUE;
    END_LOCAL;

    REPEAT i:= 1 TO SIZEOF(cl.is_case_of);
        IF (SIZEOF(cl.is_case_of[i].definition) = 1)
        THEN
            IF (NOT('ISO13584_IEC61360_DICTIONARY_SCHEMA' +
                '.ITEM_CLASS'
                IN TYPEOF(cl.is_case_of[i].definition[1])))
            THEN
                class_def_ok:= FALSE;
            END_IF;
        END_IF;
    END_REPEAT;

    RETURN(class_def_ok);

    END_FUNCTION; -- check_is_case_of_referenced_classes_definition
    (*
```

F.6.4 Определения правила стандартной схемы условных классов предметов ISO13584_IEC61360_item_class_case_of_schema

Данный подраздел определяет правило пользования стандартной схемой **ISO13584_IEC61360_item_class_case_of_schema.**

F.6.4.1 Правило видимости и применимости импортированного свойства (Imported_property_are_visible_or_applicable_rule)

Правило **imported_property_are_visible_or_applicable_rule** проверяет тот факт, что если свойство импортировано классом с помощью априорного семантического соотношения **a_priori_semantic_relationship,** то данное свойство является видимым или применимым в классе, из которого оно импортировано.

Примечание — Применимые свойства включают свойства, импортированные с помощью семантического соотношения. Данное правило дает возможность импортировать свойства из класса, куда они уже были импортированы ранее.

Пример представления на языке EXPRESS:

```
*)
RULE imported_properties_are_visible_or_applicable_rule FOR(
    a_priori_semantic_relationship, property_DET);
WHERE
    WR1: QUERY(rel <* a_priori_semantic_relationship
        | QUERY(prop <* rel.referenced_properties
        | QUERY(cl <* rel.referenced_classes
        | NOT visible_properties(cl, [prop])
        AND NOT applicable_properties(cl, [prop]))
        = rel.referenced_classes) = [ ] )
        = a_priori_semantic_relationship;
END_RULE; — imported_properties_are_visible_or_applicable_rule
(*
```

F.6.4.2 Правило видимости и применимости импортированных типов данных (Imported_**data_type**_are_visible_or_applicable_rule)

Правило **imported_data_type_are_visible_or_applicable_rule** проверяет тот факт, что если тип данных импортирован классом с помощью априорного семантического соотношения **a_priori_semantic_relationship,** то этот тип данных является видимым или применимым в классе, из которого он импортирован.

Примечание — Применимые типы данных включают типы данных, импортированные с помощью семантического соотношения. Данное правило дает возможность импортировать типы данных из класса, куда они уже были импортированы ранее.

Пример представления на языке EXPRESS:

```
*)
RULE imported_data_types_are_visible_or_applicable_rule FOR(
    a_priori_semantic_relationship, data_type_element);
WHERE
    WR1: QUERY(rel <* a_priori_semantic_relationship
        | QUERY(typ <* rel.referenced_data_types
        | QUERY(cl <* rel.referenced_classes
        | NOT visible_types(cl, [typ])
        AND NOT applicable_types(cl, [typ]))
        = rel.referenced_classes) = [ ])
        = a_priori_semantic_relationship;
END_RULE; — imported_data_types_are_visible_or_applicable_rule
(*
```

F.6.4.3 Правило использования поименованного типа (Allowed_named_type_usage_rule)

Правило **allowed_named_type_usage_rule** касается использования поименованного типа. Правило требует, чтобы только типы, применимые в классе, могли быть использованы для описания области значений свойств, объявленных в классе, с помощью атрибута **described_by**.

Пример представления на языке EXPRESS:

```
*)
RULE allowed_named_type_usage_rule FOR(class);
LOCAL
    named_type_usage_allowed: LOGICAL:= TRUE;
    is_app: LOGICAL;
    prop: property_bsu;
    cl: class;
    dtnt: SET[0:1] OF data_type_bsu:= [ ];
END_LOCAL;

REPEAT i:= 1 TO SIZEOF(class);
    cl:= class[i];
    REPEAT j:= 1 TO SIZEOF(class[i].described_by);
```

```
                prop:= cl.described_by[j];
                dtnt:= data_type_named_type(prop);

                IF (SIZEOF(dtnt) = 1) THEN
                    is_app:= applicable_types(cl.identified_by, dtnt);
                    IF (NOT is_app) THEN
                        named_type_usage_allowed:= FALSE;
                    END_IF;
                END_IF;
            END_REPEAT;
        END_REPEAT;

        WHERE
            WR1: named_type_usage_allowed;
        END_RULE; — allowed_named_type_usage_rule
        (*

        *)
    END_SCHEMA; — ISO13584_IEC61360_item_class_case_of_schema
    (*
```

F.7 Цель рассмотрения примера

В настоящем приложении приведены некоторые фрагменты физического файла, используемого для обмена данными в соответствии с МЭК 61360-DB. В приложении дан пример использования Модели языка EXPRESS (см. раздел F.3 «Стандартная словарная схема» (**ISO13584_IEC61360_dictionary_schema**)) в соответствии с ИСО 10303-21 для обмена соответствующих данных.

F.7.1 Заголовок файла

```
*/
ISO-10303-21;
HEADER;
FILE_DESCRIPTION((‘Example physical file’), ‘2;1’);
FILE_NAME(‘example.spf’, ‘2007-07-18’, (‘IEC SC3D WG2’), (),
‘Version 1’, ‘’, ‘’);
FILE_SCHEMA((‘example_schema’));
ENDSEC;
DATA;
/*
```

F.7.2 Поставщик данных

```
*/
#1=SUPPLIER_BSU(‘112/2///61360_4_1’, *);/*according to ISO 13584-26*/
#2=SUPPLIER_ELEMENT(#1, #3, ‘01’, $, $, $, #4, #5);
#3=DATES(‘1994-09-16’, ‘1994-09-16’, $);
#4=ORGANIZATION(‘IEC’, ‘IEC Maintenance Agency’, ‘The IEC Maintenance Agency’);
#5=ADDRESS(‘to be determined’, $, $, $, $, $, $, $, $, $, $);
#10=SUPPLIER_BSU(‘112/3///_00’, *);/* ISO/IEC ICS */
/*
```

F.7.3 Данные корневого класса

Корневой класс, удовлетворяющий требованиям AAA000 МЭК, задает область имен в соответствии с МЭК 61360-DB. Данный класс использует два дерева: одно для материалов, другое — для компонентов. Корневой класс определен как класс предметов **item_class.**

```
*/
#90=CLASS_BSU(‘OO’, ‘001’, #10);
#100=CLASS_BSU(‘AAA000’, ‘001’, #1);
#101=ITEM_CLASS(#100, #3, ‘01’, $, $, $, #102, TEXT(‘IEC root class that provides a name scope corresponding to IEC 61360-4-DB. It covers two trees, one for materials, one for components’), $, $, $, #90, (#110), (), (), $, (), (#110), (), $, $, $);
#102=ITEM_NAMES(LABEL(‘IEC root class’), (), LABEL(‘IEC root’), $, $);
#110=PROPERTY_BSU(‘AAE000’, ‘001’, #100);
#111=NON_DEPENDENT_P_DET(#110, #3, ‘01’, $, $,$, #112, TEXT(‘the type of tree: material or component’), $, $, $, $, (), $, $, #113, $);
#112=ITEM_NAMES(LABEL(‘type of tree’), (), LABEL(‘tree type’), $, $);
#113=NON_QUANTITATIVE_CODE_TYPE((), ‘A..8’, #114);
#114=VALUE_DOMAIN((#120,#122), $, $, (), $, $);
#120=DIC_VALUE(VALUE_CODE_TYPE(‘MATERIAL’), #121, $, $, $, $, $, $);
```

#121=ITEM_NAMES(LABEL('material tree'), (), LABEL('mat tree'), $, $);
#122=DIC_VALUE(VALUE_CODE_TYPE('COMPONS'), #123, $, $, $, $, $, $);
#123=ITEM_NAMES(LABEL('component tree'), (), LABEL('comp tree'), $, $);
/*
F.7.4 Данные о материалах
*/
#200=CLASS_BSU('AAA218', '001', #1);
#201=ITEM_CLASS(#200, #3, '01', $, $, $, #202, TEXT('root class of the materials tree'), $, $, $, #100, (#210,#230), (), (), $, (), (#210), (#205), $, 'MATERIAL', $);
#202=ITEM_NAMES(LABEL('materials root class'), (), LABEL('materials root'), $, $);
#205=CLASS_VALUE_ASSIGNMENT(#110, **String_value**('MATERIAL'));
#210=PROPERTY_BSU('AAF311', '005', #100);
#211=NON_DEPENDENT_P_DET(#210, #3, '01', $, $, $, #212, TEXT('code of the type of material'), $, $, $, $, (), $, 'A57', #213, $);
#212=ITEM_NAMES(LABEL('material type'), (), LABEL('material type'), $, $);
#213=NON_QUANTITATIVE_CODE_TYPE((), 'M..3', #214);
#214=VALUE_DOMAIN((#220,#222,#224,#226), $, $, (), $, $);
#220=DIC_VALUE(VALUE_CODE_TYPE('ACO'), #221, $, $, $, $, $, $);
#221=ITEM_NAMES(LABEL('acoustical'), (), LABEL('acoustical'), $, $);
#222=DIC_VALUE(VALUE_CODE_TYPE('MG'), #223, $, $, $, $, $, $);
#223=ITEM_NAMES(LABEL('magnetic'), (), LABEL('magnetical'), $, $);
#224=DIC_VALUE(VALUE_CODE_TYPE('OP'), #225, $, $, $, $, $, $);
#225=ITEM_NAMES(LABEL('optical'), (), LABEL('optical'), $, $);
#226=DIC_VALUE(VALUE_CODE_TYPE('TH'), #227, $, $, $, $, $, $);
#227=ITEM_NAMES(LABEL('thermal-electric'), (), LABEL('th-electric'), $, $);
#230=PROPERTY_BSU('AAF286', '005', #100);
#231=NON_DEPENDENT_P_DET(#230, #3, '01', $, $,$, #232, TEXT('The nominal density (in kg/m**3) of a material.'), $, $, $, #233, (), $, 'K02', #234, $);
#232=ITEM_NAMES(LABEL('density'), (), LABEL('density'), $, $);
#233=MATHEMATICAL_STRING('$r_d', 'ρ_d');
#234=**Real_measure_type**((), 'NR3..3.3ES2', #235, $, $, $);
#235=**Dic_unit**(#236, $);
#236=DERIVED_**Unit**((#239,#237));
#237=DERIVED_**Unit**_ELEMENT(#238, 1.);
#238=SI_**Unit**(*,.KILO.,.GRAM.);
#239=DERIVED_**Unit**_ELEMENT(#240, -3.);
#240=SI_**Unit**(*, $,.METRE.);
/*
F.7.5 Данные компонентов
*/
#300=CLASS_BSU('EEE000', '001', #1);
#301=ITEM_CLASS(#300, #3, '01', $, $, $, #302, TEXT('root class of the components tree'), $, $, $, #100, (#310,#330,#350), (), (), $, (), (#310), (#305), $, 'COMPONS', $);
#302=ITEM_NAMES(LABEL('components root class'), (), LABEL('components root'), $, $);
#305=CLASS_VALUE_ASSIGNMENT(#110, **String_value**('COMPONS'));
#310=PROPERTY_BSU('AAE001', '005', #100);
#311=NON_DEPENDENT_P_DET(#310, #3, '01', $, $, $, #312, TEXT('Code of the main functional class to which a component belongs'), $, $, $, $, (), $, 'A52', #313, $);
#312=ITEM_NAMES(LABEL('main class of component'), (), LABEL('main class'), $, $);
#313=NON_QUANTITATIVE_CODE_TYPE((), 'M..3', #314);
#314=VALUE_DOMAIN((#320,#322,#324,#326), $, $, (), $, $);
#320=DIC_VALUE(VALUE_CODE_TYPE('EE'), #321, $, $, $, $, $, $);
#321=ITEM_NAMES(LABEL('EE (electric/electronic)'), (), LABEL('EE'), $, $);
#322=DIC_VALUE(VALUE_CODE_TYPE('EM'), #323, $, $, $, $, $, $);
#323=ITEM_NAMES(LABEL('electromechanical'), (), LABEL('electromech'), $, $);
#324=DIC_VALUE(VALUE_CODE_TYPE('ME'), #325, $, $, $, $, $, $);
#325=ITEM_NAMES(LABEL('mechanical'), (), LABEL('mechanical'), $, $);
#326=DIC_VALUE(VALUE_CODE_TYPE('MP'), #327, $, $, $, $, $, $);
#327=ITEM_NAMES(LABEL('magnetic part'), (), LABEL('magnetic'), $, $);
#330=PROPERTY_BSU('AAF267', '005', #100);
#331=NON_DEPENDENT_P_DET(#330, #3, '01', $, $,$, #332, TEXT('The nominal distance (in m) between the inside of the two tapes used for taped products with axial leads'), $, $, $, #333, (), $, 'T03', #334, $);
#332=ITEM_NAMES(LABEL('inner tape spacing'), (), LABEL('inner tape spac'), $, $);
#333=MATHEMATICAL_STRING('b_tape', 'b_{tape}');

#334=LEVEL_TYPE((), (.NOM.), #335);
#335=**Real_measure_type**((), 'NR3..3.3ES2', #336, $, $, $);
#336=**Dic_unit**(#337, $);
#337=SI_**Unit**(*, $,.METRE.);
#350=PROPERTY_BSU('AAE022', '005', #100);
#351=NON_DEPENDENT_P_DET(#350, #3, '01', $, $, $,#352, TEXT('The value as specified by level (miNoMax) of the outside diameter (in m) of a component with a body of circular cross-section'), $, $, $, #353, (), $, 'T03', #354, $);
#352=ITEM_NAMES(LABEL('outside diameter'), (), LABEL('outside diam'), $, $);
#353=MATHEMATICAL_STRING('d_out', 'd_{out}');
#354=LEVEL_TYPE((), (.MIN.,.NOM.,.MAX.), #355);
#355=**Real_measure_type**((), 'NR3..3.3ES2', #356, $, $, $);
#356=**Dic_unit**(#357, #358);
#357=SI_**Unit**(*, $,.METRE.);
#358=MATHEMATICAL_STRING('m', 'm');
/*
F.7.6 Данные электрических/электронных компонентов
*/
#400=CLASS_BSU('EEE001', '001', #1);
#401=ITEM_CLASS(#400, #3, '01', $, $,$, #402, TEXT('electric/electronic components'), $, $, $, #300, (#410,#470), (), (), $, (), (#410), (#405), $, 'EE', $);
#402=ITEM_NAMES(LABEL('EE components'), (), LABEL('EE components'), $, $);
#405=CLASS_VALUE_ASSIGNMENT(#310, **String_value**('EE'));
#410=PROPERTY_BSU('AAE002', '005', #100);
#411=NON_DEPENDENT_P_DET(#410, #3, '01', $, $,$, #412, TEXT('Code of the category to which an electric/electronic component belongs.'), $, $, $, $, (), $, 'A52', #413, $);
#412=ITEM_NAMES(LABEL('category EE component'), (), LABEL('categ EE comp'), $, $);
#413=NON_QUANTITATIVE_CODE_TYPE((), 'M..3', #414);
#414=VALUE_DOMAIN((#420,#422,#424,#426,#428
,#430,#432,#434,#436,#438
,#440), $, $, (), $, $);
#420=DIC_VALUE(VALUE_CODE_TYPE('AMP'), #421, $, $, $, $, $, $);
#421=ITEM_NAMES(LABEL('amplifier'), (), LABEL('amplifier'), $, $);
#422=DIC_VALUE(VALUE_CODE_TYPE('ANT'), #423, $, $, $, $, $, $);
#423=ITEM_NAMES(LABEL('antenna (aerial)'), (), LABEL('antenna (aer)'), $, $);
#424=DIC_VALUE(VALUE_CODE_TYPE('BAT'), #425, $, $, $, $, $, $);
#425=ITEM_NAMES(LABEL('battery'), (), LABEL('battery'), $, $);
#426=DIC_VALUE(VALUE_CODE_TYPE('CAP'), #427, $, $, $, $, $, $);
#427=ITEM_NAMES(LABEL('capacitor'), (), LABEL('capacitor'), $, $);
#428=DIC_VALUE(VALUE_CODE_TYPE('CND'), #429, $, $, $, $, $, $);
#429=ITEM_NAMES(LABEL('conductor'), (), LABEL('conductor'), $, $);
#430=DIC_VALUE(VALUE_CODE_TYPE('DEL'), #431, $, $, $, $, $, $);
#431=ITEM_NAMES(LABEL('delay line'), (), LABEL('delay line'), $, $);
#432=DIC_VALUE(VALUE_CODE_TYPE('DID'), #433, $, $, $, $, $, $);
#433=ITEM_NAMES(LABEL('diode device'), (), LABEL('diode device'), $, $);
#434=DIC_VALUE(VALUE_CODE_TYPE('FIL'), #435, $, $, $, $, $, $);
#435=ITEM_NAMES(LABEL('filter'), (), LABEL('filter'), $, $);
#436=DIC_VALUE(VALUE_CODE_TYPE('IC'), #437, $, $, $, $, $, $);
#437=ITEM_NAMES(LABEL('integrated circuit'), (), LABEL('IC'), $, $);
#438=DIC_VALUE(VALUE_CODE_TYPE('IND'), #439, $, $, $, $, $, $);
#439=ITEM_NAMES(LABEL('inductor'), (), LABEL('inductor'), $, $);
#440=DIC_VALUE(VALUE_CODE_TYPE('LAM'), #441, $, $, $, $, $, $);
#441=ITEM_NAMES(LABEL('lamp'), (), LABEL('lamp'), $, $);
#470=PROPERTY_BSU('AAE754', '005', #100);
#471=NON_DEPENDENT_P_DET(#470, #3, '01', $, $, $,#472, TEXT('The number of electrical terminals of an electric/electronic or electromechanical component'), $, $, $, #473, (), $, 'Q56', #474, $);
#472=ITEM_NAMES(LABEL('number of terminals'), (LABEL('number of pins')), LABEL('nr of terminals'), $, $);
#473=MATHEMATICAL_STRING('N_term', 'N_{term}');
#474=INT_TYPE((), 'NR1..4');
ENDSEC;
END—ISO-10303-21;

Приложение G
(справочное)

Обзор основных классов и категорий свойств

Таблица G.1 — Обзор основных классов и категорий свойств

Категория/подкатегория	Основной класс	Описание/предмет рассмотрения
Неколичественные типы элементов данных	A	Идентификации и индикаторы
Количественные типы элементов данных	C	Физическая химия и молекулярная физика
	E	Электричество и магнетизм
Физические меры:	F	Периодические и родственные явления
Нефизические меры:	G	Акустика
Финансовые меры	H	Теплота
Прочие меры	J	Информация
	K	Механика
	L	Свет и родств. электромагн. излучение
	T	Пространство и время
	U	Атомная и ядерная физика
	V	Ядерные реакции и ионизир. излучение
	W	Физика твердого тела
	M	Финансовое исчисление
	P	Финансовые величины: цены, тарифы
	Q	Счетные величины, расчеты
	R	Бизнес-соотношения, процентные соотношения

Приложение Н
(справочное)

Обзор кодов классификации типов количественных типов элементов данных

Здесь представлена полная классификация и названия основных классов. Коды основных классов идут в алфавитном порядке.

C Физическая химия и молекулярная физика [ИСО 80000-9 (ранее ИСО 31-8)**]**

Коды классификации типов отсутствуют.

E Электрические и магнитные величины [МЭК 80000-6 (ранее ИСО 31-5)**]**

E01	электрический ток	А
E02	электрический заряд, электрические величины	К
E03	объемный заряд, объемная плотность заряда, зарядная плотность	$К/м^3$
E04	атмосферный заряд, поверхностная плотность заряда	$К/м^2$
E05	напряженность электрического поля	В/м
E06	электрический потенциал, разность потенциалов напряжение, электродвижущая сила	В
E07	плотность электрического потока	$К/м^2$
E08	электрический поток	К
E09	электроемкость	Ф
E10	проницаемость, электрическая константа, проницаемость вакуума	Ф/м
E11	относительная проницаемость	1
E12	электрическая восприимчивость	1
E13	электрическая поляризация	$К/м^2$
E14	момент электрического диполя	К·м
E15	атмосферный электрический ток, плотность электрического тока	$А/м^2$
E16	линейный электрический ток, плотность линейного электрического тока	А/м
E17	напряженность магнитного поля	А/м
E18	разность магнитных потенциалов, магнитодвижущая сила, связывающий ток	А
E19	плотность магнитного потока, магнитная индукция	Т
E20	магнитный поток	Вб
E21	векторный магнитный потенциал	Вб/м
E22	самоиндукция, взаимная индукция	В
E23	фактор сопряжения, фактор утечки	1
E24	проницаемость, магнитная константа, проницаемость вакуума	Г/м
E25	относительная проницаемость	1
E26	магнитная восприимчивость	1
E27	магнитный момент, электромагнитный момент	$А·м^2$
E28	намагничивание	А/м
E29	магнитная поляризация	Т
E30	объемная электромагнитная энергия, плотность электромагнитной энергии	$Дж/м^3$
E31	вектор Пойнтинга	$Вт/м^2$
E32	Не используется	
E33	сопротивление (постоянному току)	Ом
E34	проводимость (постоянного тока)	Сим
E35	мощность (постоянного тока)	Вт
E36	удельное сопротивление	Ом/м
E37	удельная проводимость	Сим/м
E38	магнитное сопротивление	$Генри^{-1}$
E39	магнитная проводимость	Генри
E40	Не используется	
E41	частота, частота вращения	Гц, $сек^{-1}$
E42	угловая частота, угловая частота переменного тока	рад/сек, $сек^{-1}$
E43	разность фаз	рад
E44	импеданс, модуль импеданса, сопротивление, реактивное сопротивление	Ом
E45	полная проводимость, модуль	Сим
E46	фактор качества	1
E47	фактор потерь	1
E48	угол потерь	рад
E49	активная мощность	Вт
E50	кажущаяся мощность, реактивная мощность	В·А
E51	фактор мощности	1
E52	активная энергия	Дж, Вт·час

F Величины периодических и связанных явлений [ИСО 80000-3 (ранее ИСО 31-2)**]**

F01	период, время повторения	сек	F08	фазовая скорость, групповая скорость	м/с
F02	константа времени для величины, изменяющейся по экспоненте	сек	F09	уровень интенсивности поля	Нп, Б
F03	частота, частота вращения	Гц, $с^{-1}$	F10	уровень мощности	Нп, Б
F04	угловая частота, угловая скорость	рад/с, $сек^{-1}$	F11	коэффициент демпфирования	$с^{-1}$, Нп/с
F05	длина волны	м	F12	логарифмический декремент	Нп
F06	волновое число, повторяемость	$м^{-1}$	F13	коэффициент затухания, фазовый коэффициент, коэффициент распространения	$м^{-1}$
F07	угловое волновое число, угловая повторяемость	рад/м $м^{-1}$			

G Акустические величины [ИСО 80000-8 (ранее ИСО 31-7)**]**

G01	период, время повторения	сек	G19	механический импеданс	Н·с/м
G02	частота	Гц	G20	характеристический импеданс среды, поверхностная плотность механического импеданса	Па·с/м
G03	частотный интервал	октава			
G04	угловая частота, угловая скорость	рад/с, $с^{-1}$			
G05	длина волны	м		механический импеданс	Н·с/м
G06	волновое число, повторяемость	$м^{-1}$	G21	уровень звукового давления	Б
G07	угловое волновое число	рад/м	G22	уровень звуковой мощности	Б
	угловая повторяемость	$м^{-1}$	G23	коэффициент демпфирования	$с^{-1}$, Нп/с
G08	объемная масса, массовая плотность, плотность	$кг/м^3$	G24	константа времени, время релаксации	с
			G25	логарифмический декремент	Нп
G09	статическое давление, звуковое давление	Па	G26	коэффициент затухания, фазовый коэффициент, коэффициент распространения	$м^{-1}$
G10	смещение звуковой частицы	м			
G11	скорость звуковой частицы	м/с	G27	фактор диссипации, фактор отражения, фактор передачи, фактор поглощения	1
G12	ускорение звуковой частицы	$м/с^2$			
G13	объемный расход	$м^3/с$			
G14	скорость звука (фазовая скорость), групповая скорость	м/с	G28	индекс редукции звука	Б
G15	плотность звуковой энергии, объемная звуковая энергия	$Дж/м^3$	G29	эквивалентная область поглощения поверхности или объекта	$м^2$
G16	звуковая мощность	Вт	G30	время реверберации	с
G17	звуковая интенсивность	$Вт/м^2$	G31	уровень громкости	фон
G18	акустический импеданс	$Па·с/м^3$	G32	громкость	сон

H Тепловые величины [ИСО 80000-5 (ранее ИСО 31-4)**]**

H01	термодинамическая температура	K	H10	коэффициент теплопередачи, поверхностный коэффициент теплопередачи	$Вт/(м^2.K)$
H02	температура по Цельсию	°C			
H03	коэффициент линейного расширения, коэффициент кубического расширения, относительный коэффициент давления	K^{-1}	H11	сопротивление теплоизоляции, коэффициент теплоизоляции	$м^2·K/Вт$
			H12	тепловое сопротивление	K/Вт
H04	коэффициент давления	Па/K	H13	тепловая проводимость	Вт/K
H05	изотермическая сжимаемость, изэнтропическая сжимаемость	$Па^{-1}$	H14	тепловая диффузность	$м^2/с$
			H15	теплоемкость	Дж/K
H06	количество теплоты, теплота	Дж	H16	массовая теплоемкость, удельная теплоемкость при: постоянном давлении, постоянном объеме, насыщении	Дж/(кг·K)
H07	тепловой расход	Вт			
H08	атмосферный тепловой расход, плотность теплового расхода	$Вт/м^2$			
H09	теплопроводность	Вт/(м.K)			

H17	относительная массовая теплоемкость относительная удельная теплоемкость изэнтропическая экспонента	1		удельная термодинамическая энергия, массовая энтальпия, удельная энтальпия, массовая свободная энергия Гельмгольца,	
H18	энтропия	Дж/К			
H19	массовая энтропия, удельная энтропия	Дж/(кг·К)		удельная свободная энергия Гельмгольца,	
H20	энергия, термодинамическая энергия, энтальпия, свободная энергия Гельмгольца, функция Гельмгольца, свободная энергия Гиббса, функция Гиббса	Дж		удельная функция Гельмгольца, массовая свободная энергия Гиббса, удельная свободная энергия Гиббса, удельная функция Гиббса	
H21	массовая энергия, удельная энергия, массовая термодинамическая энергия,	ДЖ/кг	H22	функция Масью	Дж/К
			H23	Функция Планка	Дж/К

J Величины информации

J01	длина слова	бит	J04	объемная плотность хранения	бит/м^3
	мера памяти	байт	J05	скорость передачи	бит/с, Бд
	длина регистра	слово			
J02	линейная плотность хранения	бит/м	J06	частота появления ошибок, скорость кодирования эффективность	1
J03	поверхностная плотность хранения	бит/м^2			

K Величины механики [ИСО 80000-4 (ранее ИСО 31-3)**]**

K01	масса	кг		удлинение), сдвиговая деформация, объемная деформация (общая деформация)	
K02	объемная масса, массовая плотность, плотность	кг/м^3			
K03	относительная объемная масса, относительная массовая плотность, относительная плотность	1	K17	отношение Пуассона, число Пуассона	1
			K18	модуль упругости, модуль сдвига модуль жесткости, общий модуль, модуль сжатия	Па
K04	массовый объем, удельный объем	м^3/кг			
K05	линейная масса, линейная плотность	кг/м			
K06	атмосферная масса, поверхностная плотность	кг/м^2	K19	сжимаемость (общая сжимаемость)	Па$^{-1}$
K07	момент инерции	кг·м^2	K20	второй момент области (второй осевой момент	м^4
K08	количество движения	кг·м/с			
K09	сила, вес	Н	K21	момент сопротивления сечения	м^3
K10	импульс	Н·с	K22	динамический фактор трения, статический фактор трения	1
K11	момент количества движения, угловое количество движения	кг·м^2/с			
			K23	вязкость (динамическая вязкость)	Па·сек
K12	момент силы, момент пары, крутящий момент	Н·м	K24	кинематическая вязкость	м^2/с
			K25	поверхностное натяжение	Н/м
K13	угловой импульс	Н·м·с	K26	энергия, работа, потенциальная энергия, кинетическая энергия	Дж
K14	графитационная постоянная область), второй полярный момент области	Н·м^2/кг^2			
			K27	мощность	Вт
K15	давление, нормальное напряжение, сдвиговое напряжение	Па	K28	К.п.д.	1
			K29	массовый расход	кг/с
K16	линейная деформация (относительное	1	K30	объемный расход	м^3/с

L	**Величины света и электромагнитного излучения [ИСО 80000-7 (ранее ИСО 31-6)]**	
L01	частота	Гц
L02	угловая частота	рад/с, $с^{-1}$,
L03	длина волны	м
L04	повторяемость, волновое число	$м^{-1}$
L05	угловая повторяемость, угловое волновое число	рад/м, $м^{-1}$
L06	не используется	
L07	лучистая энергия	Дж
L08	плотность лучистой энергии	$Дж/м^3$
L09	спектральная концентрация плотности лучистой энергии	$Дж/м^4$
L10	лучистая мощность, поток лучистой энергии	Вт
L11	флюенс лучистой энергии	$Дж/м^2$
L12	интенсивность флюенса лучистой энергии	$Вт/м^2$
L13	лучистая интенсивность	Вт/ср
L14	энергетическая яркость	Вт/(ср. $м^2$)
L15	лучистая светимость	$Вт/м^2$
L16	облученность	$Вт/м^2$
L17	лучистая экспозиция	$Дж/м^2$
L18	не используется	
L19	не используется	
L20	не используется	
L21	излучательная способность, спектральная излучательная способность, спектральная излучательная способность по направлению	1
L22	число фотонов	1
L23	поток фотонов	$с^{-1}$
L24	интенсивность потока фотонов	$с^{-1}$/ср
L25	яркость фотона энергетическая яркость фотона	$с^{-1}$/(ср·$м^2$)
L26	светимость фотона	$с^{-1}/м^2$
L27	облученность фотона	$с^{-1}/м^2$
L28	экспозиция фотона	$м^{-2}$
L29	интенсивность света	кд
L30	световой поток	лм
L31	количество света	лм·с
L32	яркость	кд/м
L33	светимость (в точке поверхности)	$лм/м^2$
L34	Освещенность	лк
L35	оптическое экспонирование	лк.с
L36	сила света, спектральная сила света, максимальная спектральная сила света	лм/Вт
L37	световая эффективность, спектральная световая эффективность	1
L38	колориметрические функции МКО	1
L39	трихроматические координаты	1
L40	фактор спектрального поглощения, фактор спектрального отражения, фактор спектральной передачи, фактор спектральной энергетической яркости	1
L41	оптическая плотность	1
L42	коэффициент линейного затухания, коэффициент линейного поглощения	$м^{-1}$
L43	молярный коэффициент поглощения	$м^2$/моль
L44	показатель преломления	1
L45	расстояние до объекта, расстояние до изображения, фокусное расстояние	м
L46	вергенция, мощность линзы	$м^{-1}$

M	**Вспомогательные величины**	
M51	количество товаров в ситуациях	
M52	количество товаров в транзакциях	
M53	количество товаров в агрегациях	
M62	количество информационных объектов в транзакциях	

P	**Цены, тарифы**
P51	цены товаров в ситуациях
P52	цены товаров в транзакциях

Q	**Безразмерные бизнес-величины и оценки**		
Q31	количество определенных периодов времени в ситуациях	Q56	количество функций продуктов
Q33	количество совокупных периодов времени	Q59	количество продуктов в статистике качества
Q51	количество товаров в ситуациях	Q61	количество документов (частей документации)
Q52	количество товаров в транзакциях	Q62	количество информационных объектов в транзакциях
Q53	количество товаров в агрегациях	Q63	количество информационных объектов в агрегациях
Q54	количество товаров в иерархических структурах		

R **Бизнес-параметры и процентные соотношения**

R41	количество соотношений (процентных) в отношении конкретных лиц в ситуациях	R53	количество соотношений (процентных) в отношении товаров в агрегациях
R51	количество соотношений (процентных) в отношении товаров в ситуациях	R54	количество соотношений (процентных) в отношении товаров в иерархических структурах
R52	количество соотношений (процентных) в отношении товаров в транзакциях		

T **Величины пространства и времени [ИСО 80000-3** (ранее ИСО 31-1)**]**

T01	угол	рад, °, ', "	T07	время, интервал времени, продолжительность	сек, мин, час, сут
T02	пространственный угол	ср			
T03	длина, ширина, высота, толщина,радиус, диаметр, длина путь, расстояние, декартовы координаты,	м	T08	угловая скорость	рад/с
			T09	угловое ускорение	рад/$с^2$
			T10	скорость	м/с
T04	радиус кривизны, кривизна	$м^{-1}$	T11	ускорение, ускорение свободного падения, ускорение под действием силы тяжести	м/$с^2$
T05	область	$м^2$			
T06	объем	$м^3$			

U **Величины атомной и ядерной физики [ИСО 80000-10** (ранее ИСО 31-9)**]**

Коды классификации типов отсутствуют.

V **Величины ядерных реакций и ионизирующей радиации [ИСО 80000-10** (ранее ИСО 31-10)**]**

V01	энергия реакции	Дж, эВ	V21	средний линейный диапазон значений масс	
V02	энергия резонанса	Дж, эВ			
V03	поперечное сечение, суммарное поперечное сечение	$м^2$	V22	средний массовый диапазон	кг/$м^2$
			V23	линейная ионизация одной частицей	m^{-1}
V04	угловое поперечное сечение	$м^2$/ср			
V05	спектральное поперечное сечение	$м^2$/Дж	V24	суммарная ионизация частицей	1
V06	спектральное угловое поперечное сечение	$м^2$/(ср. Дж)	V25	средняя потеря энергии, приходящаяся на пару образованных ионов	ДЖ, эВ
V07	макроскопическое поперечное сечение, объемное поперечное сечение	$м^{-1}$			
			V26	мобильность	$m^2/(V·s)$
V08	поток частиц	$м^{-2}$	V27	плотность числа ионов, плотность ионов	$м^{-3}$
V09	интенсивность потока частиц (плотность потока частиц)	$м^{-2}·с^{-1}$	V28	коэффициент рекомбинации	$м^3$/с
			V29	плотность числа нейтронов	$м^{-3}$
V10	поток энергии	Дж/$м^2$	V30	скорость нейтронов	м/с
V11	интерсивность потока энергии (плотность потока энергии)	Вт/$м^2$	V31	интенсивность потока нейтронов (плотность потока нейтронов)	$с^{-1}·м^{-2}$
V12	мгновенная плотность частиц	$м^{-2}·с^{-1}$			
V13	коэффициент линейного затухания	$м^{-1}$	V32	коэффициент диффузии для плотности числа нейтронов	$м^2$/с
V14	коэффициент массового затухания	$м^2$/кг			
V15	коэффициент молярного затухания	$м^2$/моль	V33	коэффициент диффузии для плотности потока нейтронов	м
V16	коэффициент атомного затухания	$м^2$			
V17	полутолщина, половинное значение толщины	м	V34	плотность источников нейтронов	$с^{-1}/м^3$
			V35	падающая плотность	$м^{-3}$/с
V18	суммарная линейная мощность торможения	Дж/м, эВ/м	V36	вероятность ухода от резонанса	1
			V37	летаргия нейтронов в процессе замедления	1
V19	суммарная атомная мощность торможения	$J·m^2$, $эВ·m^2$	V38	средняя логарифмическая энергия декремент	1
V20	суммарная массовая мощность торможения	$J·m^2$/кг, $эВ·m^2$/кг			

V39	средняя длина свободного пробега	м
V40	область торможения, область диффузии, область миграции	$м^2$
V41	тормозной путь, путь диффузии, путь миграции	м
V42	выход нейтронов на каждое расщепление, выход нейтронов на каждое поглощение	1
V43	фактор быстроты расщепления	1
V44	фактор использования тепла	1
V45	вероятность отсутствия утечек	1
V46	мультипликативный фактор, мультипликация бесконечной среды	1
V47	фактор реактивности, фактор эффективной мультипликативности	1
V48	константа времени реактора	с
V49	активность	Бк
V50	сообщенная энергия, средняя сообщенная энергия	Дж
V51	удельная энергия поглощения, поглощенная доза	Gy
V52	эквивалент дозы	Sv(ДЖ/кг)
V53	интенсивность поглощения дозы	Gy/с
V54	линейная передача энергии	ДЖ/м, эВ/м
V55	керма	Gy
V56	интенсивность кермы	Gy/с
V57	коэффициент передачи энергии массы	$м^2$/кг
V58	экспозиция	С/кг, R
V59	интенсивность экспозиции	С/(кг·с)

W Величины физики твердого тела

Коды классификации типов отсутствуют.

Приложение I
(справочное)

Диаграмма EXPRESS-G

Данное приложение содержит диаграммы EXPRESS-G(см. приложение F). Понятие диаграммы EXPRESS-G определено в приложении А ИСО 10303-11:2004.

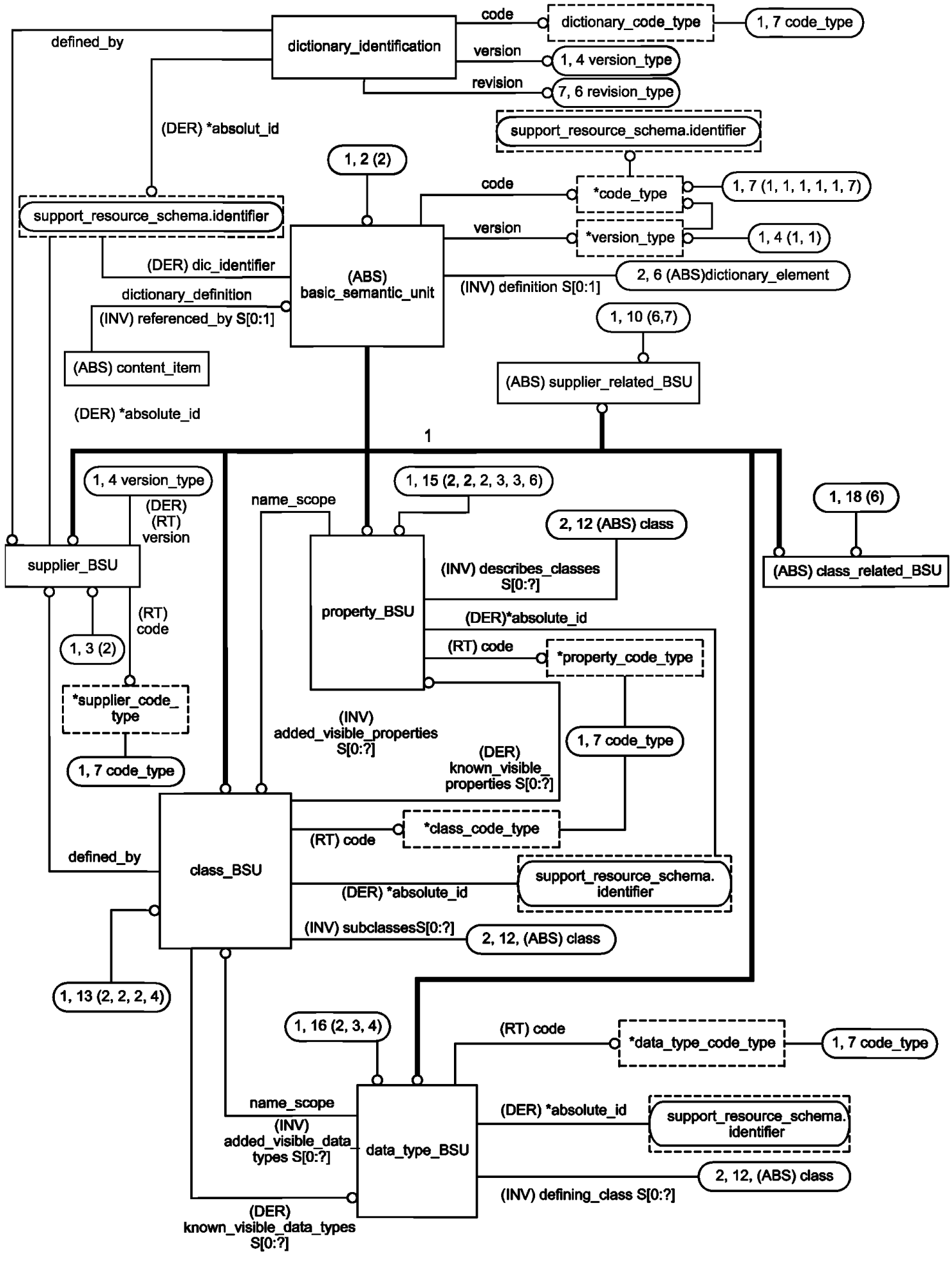


Рисунок I.1 — Диаграмма EXPRESS-G № 1 (из семи) стандартной словарной схемы **ISO13584_IEC61360_ dictionary_schema**

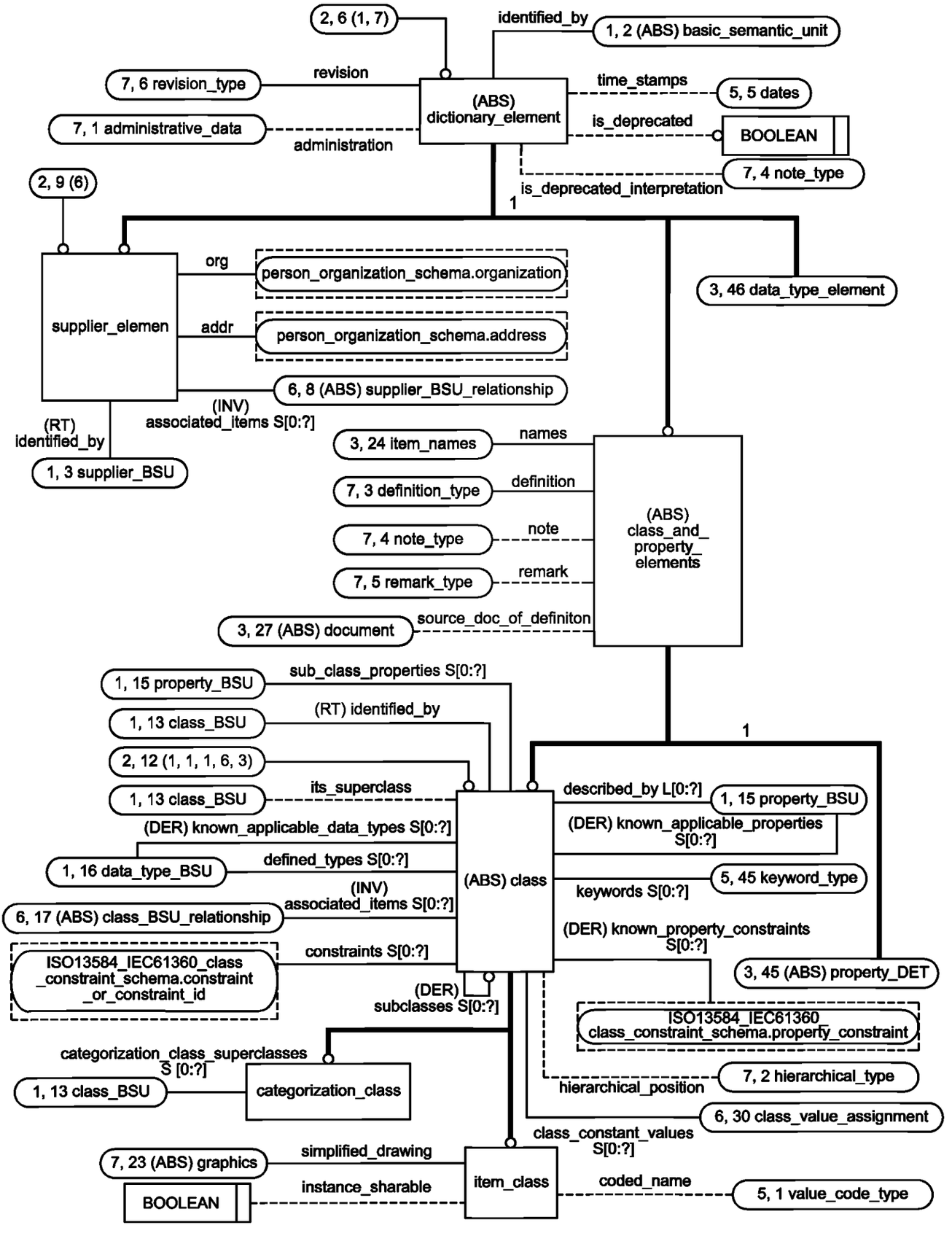


Рисунок I.2 — Диаграмма EXPRESS-G №2 (из семи) стандартной словарной схемы **ISO13584_IEC61360_ dictionary_schema**

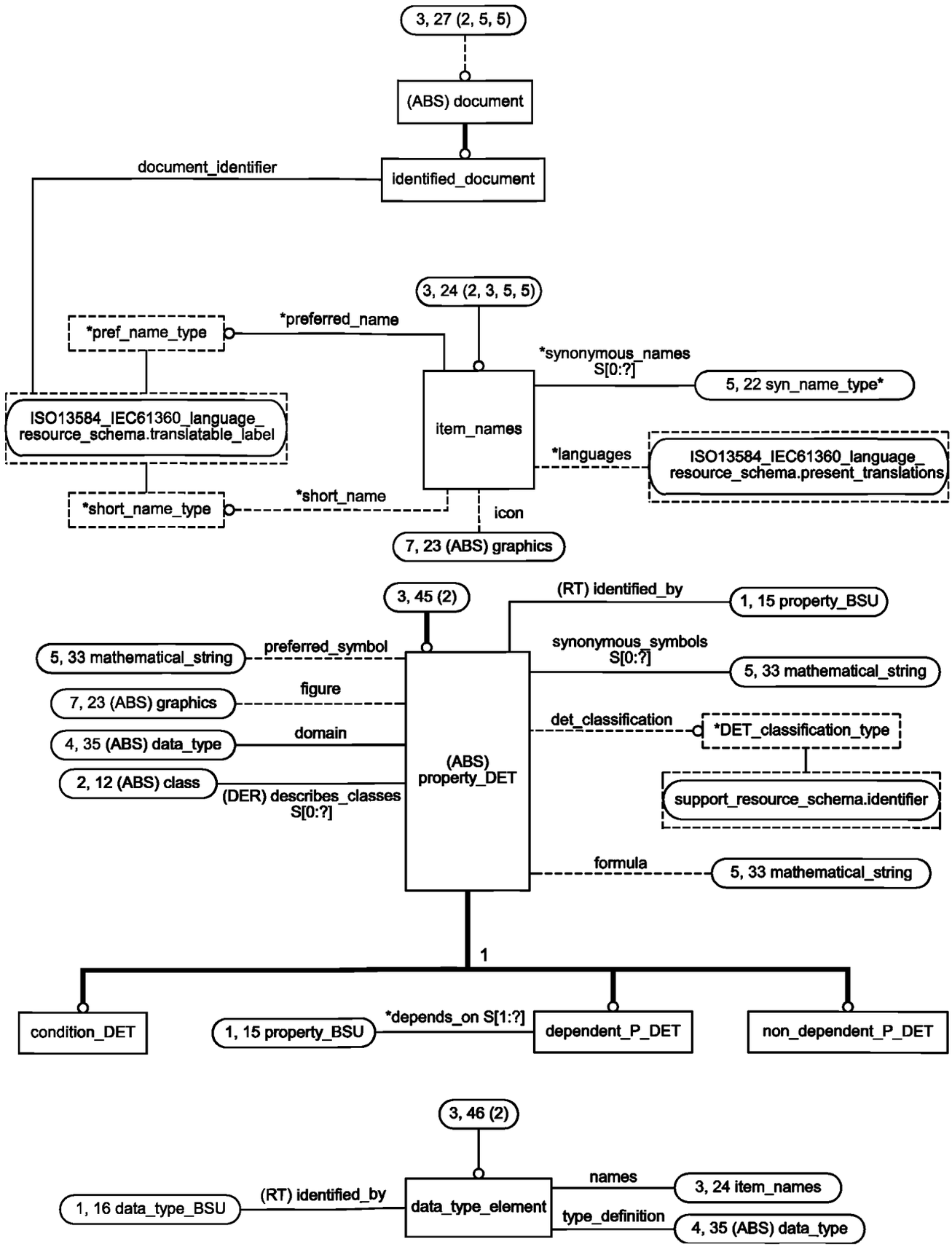


Рисунок I.3 — Диаграмма EXPRESS-G №3 (из семи) для стандартной словарной схемы **ISO13584_IEC61360_ dictionary_schema**

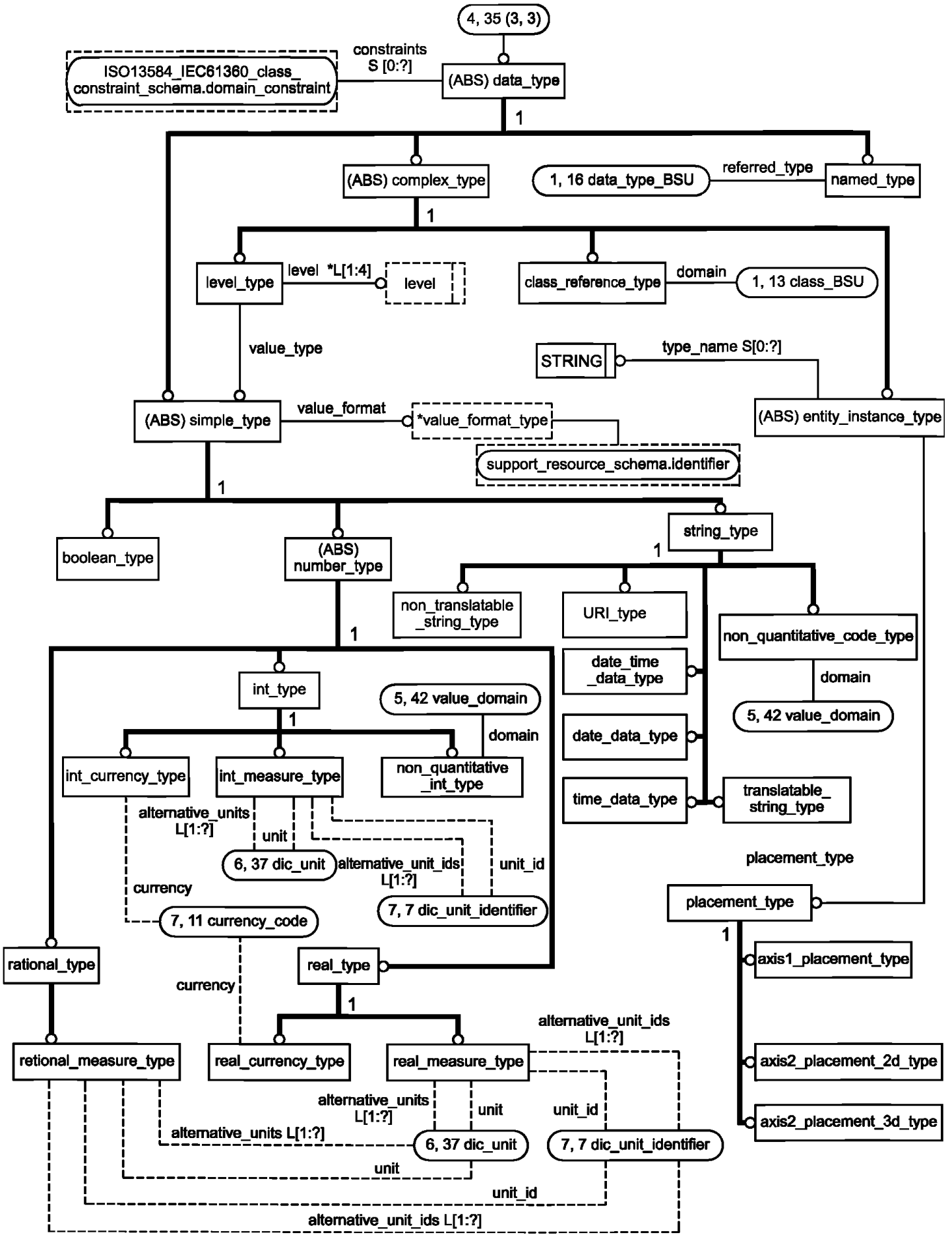


Рисунок I.4 — Диаграмма EXPRESS-G №4 (из семи) для стандартной словарной схемы **ISO13584_IEC61360_ dictionary_schema**

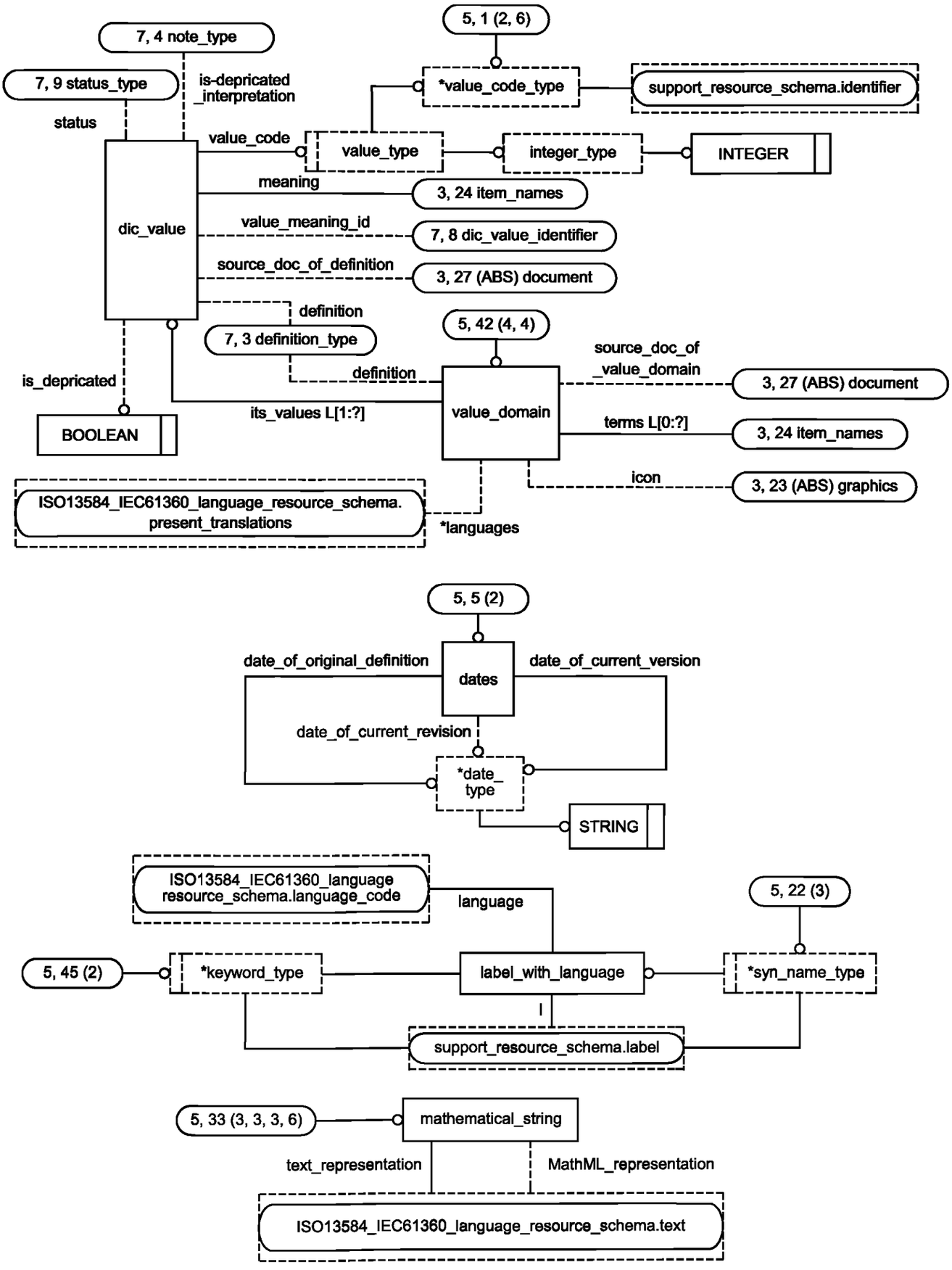


Рисунок I.5 — Диаграмма EXPRESS-G №5 (из семи) для стандартной словарной схемы **ISO13584_IEC61360_ dictionary_schema**

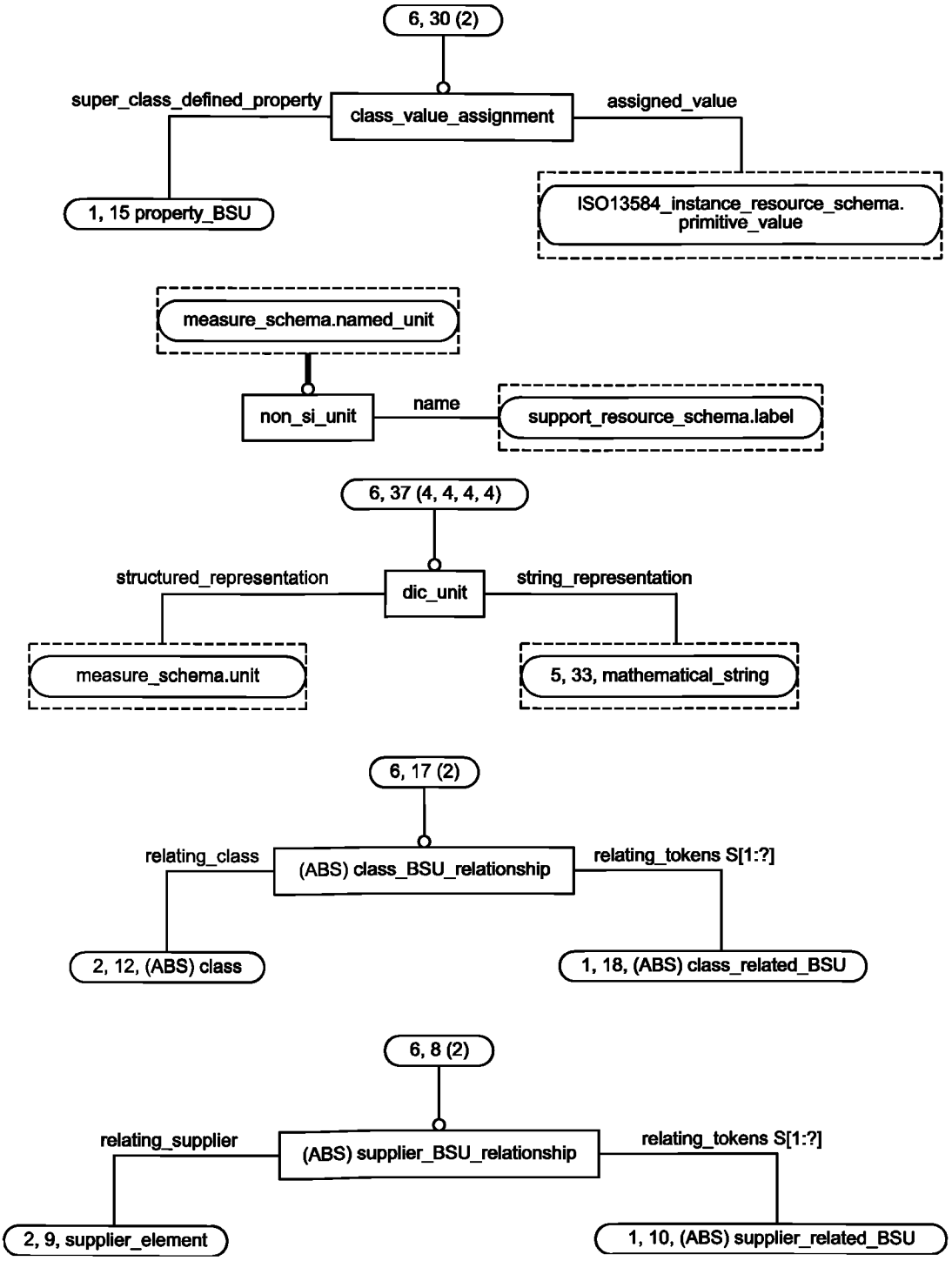


Рисунок I.6 — Диаграмма EXPRESS-G №6 (из семи) для стандартной словарной схемы **ISO13584_IEC61360_ dictionary_schema**

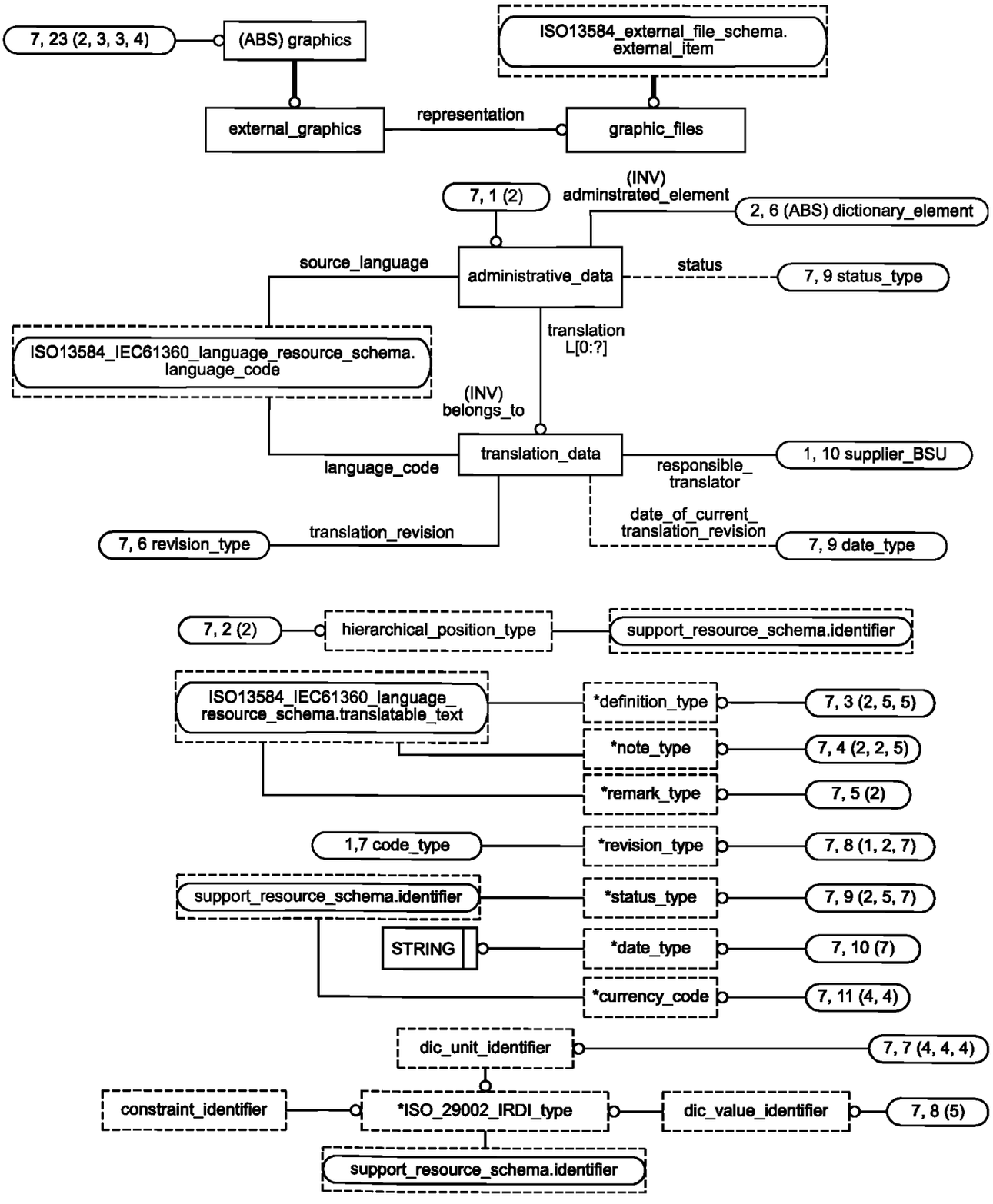


Рисунок I.7 — Диаграмма EXPRESS-G №7 (из семи) для стандартной словарной схемы **ISO13584_IEC61360_ dictionary_schema**

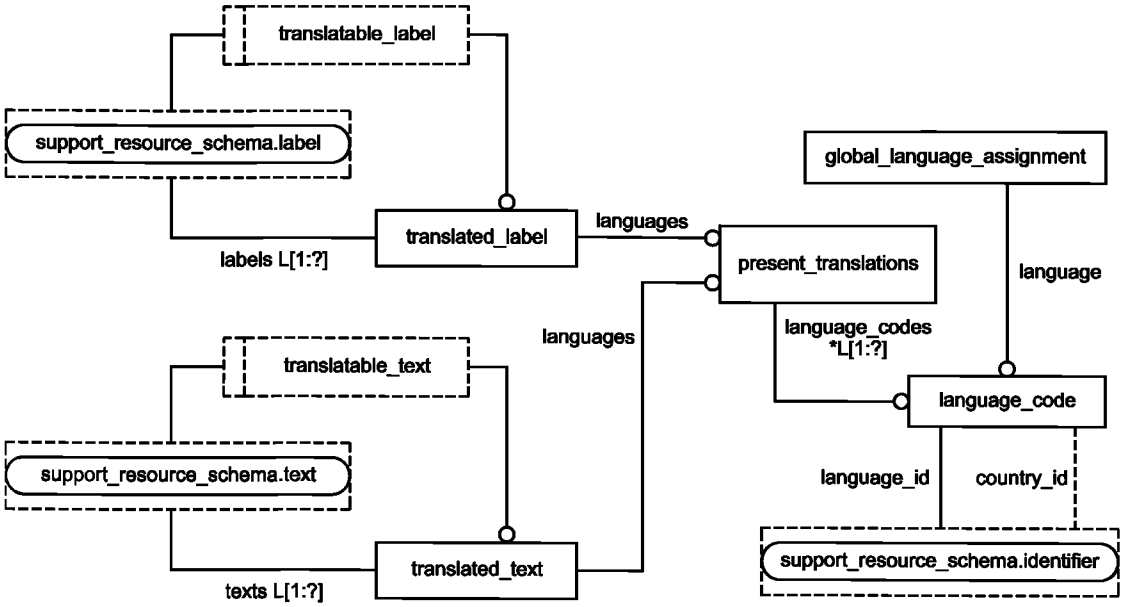


Рисунок I.8 — Диаграмма EXPRESS-G № 1 (единственная) стандартной схемы языкового ресурса **ISO13584_ IEC61360_language_resource_schema**

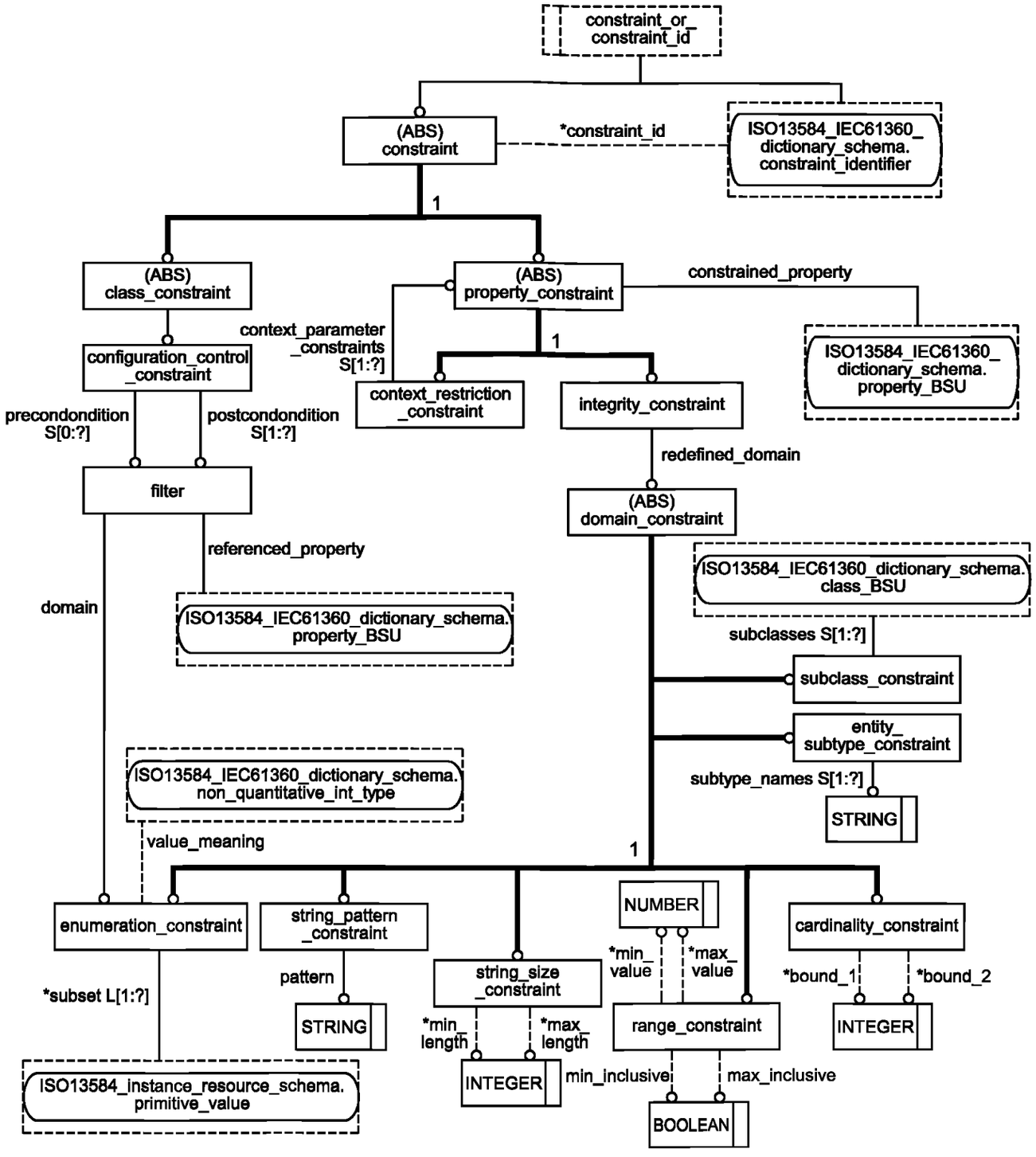


Рисунок I.9 — Диаграмма EXPRESS-G №1 (единственная) стандартной схемы ограничений **ISO13584_IEC61360_ constraint_schema**

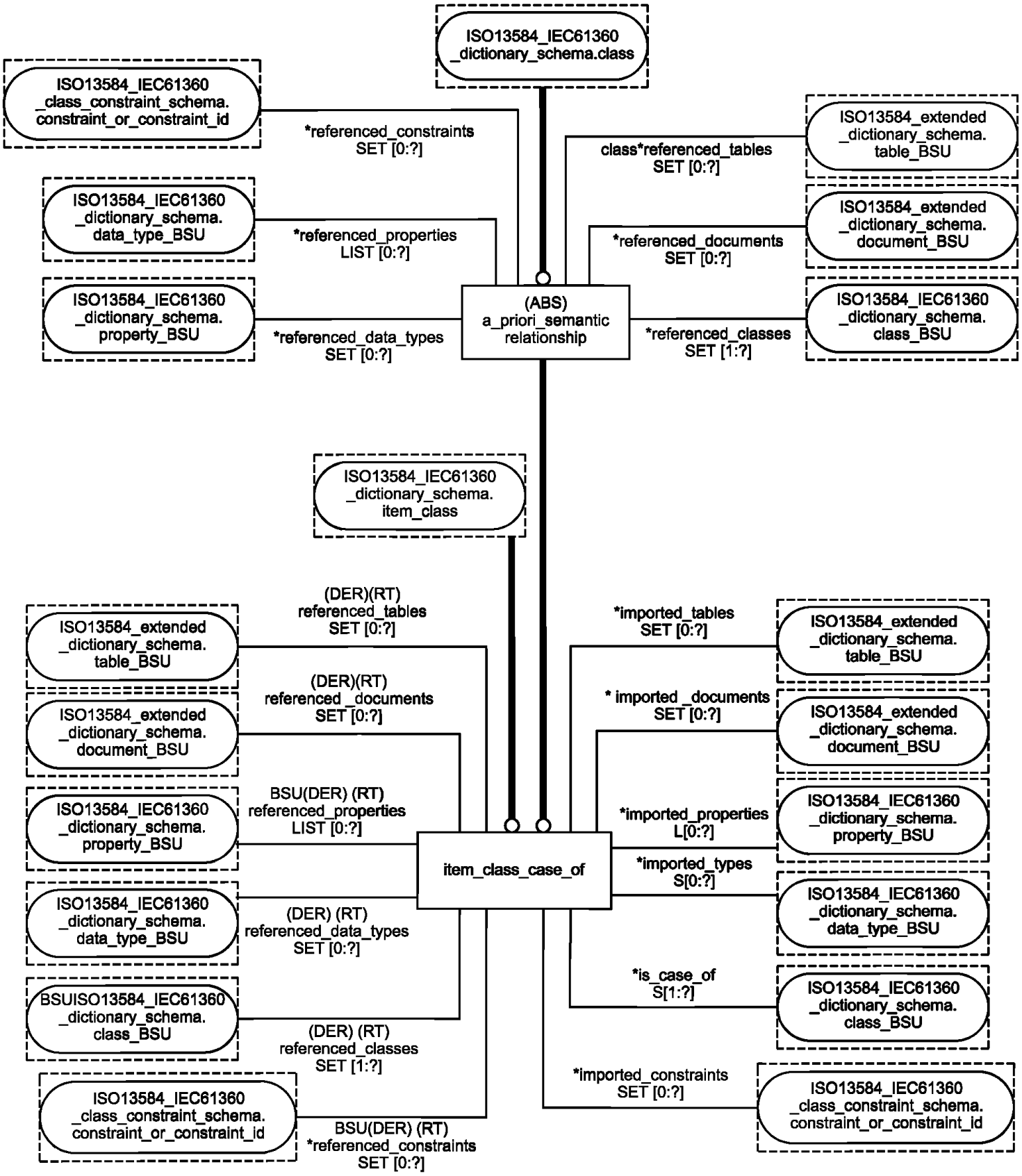


Рисунок I.10 — Диаграмма EXPRESS-G № 1 (единственная) для стандартной схемы условного класса продуктов **ISO13584_IEC61360_item_class_case_of_schema**

Приложение J
(справочное)

Частные словари

Модель данных EXPRESS, опубликованная в настоящем стандарте и дублированная (для удобства) в ИСО 13584-42, допускает описание словарей, составленных из классов, свойств и типов данных. Данная модель обеспечивает их уникальную идентификацию с помощью механизма Базовой Семантической Единицы (БСЕ). Рассматриваемая модель данных допускает описание иерархий классов, структурированных в соответствии со структурой дерева, с помощью простого механизма наследственности. Указанную модель данных используют только отдельные автономные словари.

Использование рассматриваемой модели данных способствует разработке нескольких словарей. Во время разработки словаря может оказаться необходимым сослаться на конкретный класс, свойство или тип данных, определенные ранее в другом словаре. Возможность импортирования свойств и типов данных в разрабатываемый словарь обеспечивается некоторым условным соотношением, которое может быть использовано совместно с полной общей Словарной моделью ИСО 13584/МЭК 61360, задокументированной как в ИСО 13584-25, так и в МЭК 61360-5. Более того, данное соотношение допускает импортирование внешне определенных свойств и типов данных, обеспечивающих поддержку частных словарей и исключение дублирования словарей. При этом каждый словарь может определять свою собственную структуру класса.

Полная общая Словарная модель ИСО 13584/МЭК 61360 предлагает два механизма:

- сущность языка EXPRESS **a_priori_case_of_semantic_relationship** (априорное условное семантическое соотношение) допускает прямое использование свойств или типов данных, определенных во внешнем словаре (словарях), без их повторного описания;
- сущность языка EXPRESS **a_posteriori_case_of_relationship** (апостериорное условное соотношение) допускает (если некоторые свойства или типы данных уже определены в разрабатываемом словаре) их отображение на соответствующие свойства или типы данных, определенные во внешнем словаре.

Указанные механизмы позволяют разрабатывать словари, ссылающиеся на элементы данных, определенные в других словарях, без изменения их семантического смысла. Более того, условные соотношения заносятся в словари и используются для автоматической интеграции словарей, основанных на указанных стандартных словарях.

Данный механизм также может быть использован при разработке словаря конечного пользователя. Как правило, конечному пользователю словаря не нужна вся структура класса, определенная в стандартных словарях. Ему часто достаточно иметь возможность обмениваться информацией с другими пользователями, словари которых основаны на том же стандартном словаре (словарях). Если конечный пользователь определяет свою собственную иерархию, но отображает каждый свой класс (с помощью условного соотношения) на соответствующий стандартный класс, и если пользователь импортирует все существующие стандартные свойства, необходимые в текущем контексте, и добавляет свои собственные особые свойства, то он может приспособить свой собственный словарь для обмена стандартной информацией с другими пользователями. Рассматриваемый подход к разработке частных словарей конечных пользователей — это суть предложения, содержащегося в настоящем стандарте.

Приложение К
(справочное)

Информационная поддержка практической реализации

Для поддержки практической реализации приложения можно получить дополнительную информацию. Если информация существует, то ее можно найти по следующему адресу (URL):

gost.e-ecolog.ru

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО 8601:2004	IDT	ГОСТ ИСО 8601:2001 Система стандартов по информации, библиотечному и издательскому делу. Представление дат и времени. Общие требования
ИСО 10303-11:2004	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
МЭК 61360-2:2012	—	*

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

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

IDT — идентичный стандарт.

Библиография

[1] ИСО 639-1:2002 (ISO 639-1:2002) — Коды для представления названий языков. Часть 1. Двухбуквенный код (Codes for the representation of names of languages. Part 1. Alpha-2 code)

[2] ИСО 639-2:2002 (ISO 639-2:2002) — Коды для представления названий языков. Часть 2. Трехбуквенный код (Codes for the representation of names of languages — Part 2: Alpha-3 code)

[3] ИСО 704:2009 (ISO 704:2009) — Терминологическая деятельность. Принципы и методы (Terminology work — Principles and methods)

[4] ИСО 843:1997 (ISO 843:1997) — Информация и документация. Транслитерация и транскрипция букв греческого алфавита буквами латинского алфавита (Information and documentation — Conversion of Greek characters into Latin characters)

[5] ИСО 1087-1:2000 (ISO 1087-1:2000) — Терминологическая работа. Словарь. Часть 1. Теория и применение (Terminology work. Vocabulary. Part 1. Theory and application)

[6] ИСО 3166-1:2006 (ISO 3166-1:2006) — Коды для представления названий стран и единиц их административно-территориального деления. Часть 1. Коды стран (Codes for the representation of names of countries and their subdivisions — Part 1: Country codes)

[7] ИСО 4217:2008 (ISO 4217:2008) — Коды для представления валют и фондов (Codes for the representation of currencies and funds)

[8] ИСО/МЭК 8824-1:2008 (ISO/IEC 8824-1:2008) — Информационные технологии. Нотация абстрактного синтаксиса версии 1 (ASN.1). Часть 1. Спецификация базовой нотации (Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation)

[9] ИСО/МЭК 8859-1:1998 (ISO/IEC 8859-1:1998) — Информационные технологии. 8-битовые однобайтовые наборы кодированных графических знаков. Часть 1. Латинский алфавит №1 (Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1)

[10] ИСО 10303-1:1994 (ISO 10303-1:1994) — Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 1. Обзор и основные принципы (Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles)

[11] ИСО 10303-21:2002 (ISO 10303-21:2002) — Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 21. Методы реализации. Кодирование открытого текста структуры обмена (Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure)

[12] ИСО 10303-41:2005 (ISO 10303-41:2005) — Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 41. Интегрированные родовые ресурсы. Основы описания продукции и программного обеспечения (Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support)

[13] ИСО 10303-42:2003 (ISO 10303-42:2003) — Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 42. Интегрированные родовые ресурсы. Геометрическое и топологическое представление (Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation)

[14] ИСО/МЭК 11179-5:2005 (ISO/IEC 11179-5:2005) — Информационные технологии. Реестры метаданных (MDR). Часть 5. Принципы присвоения имен и идентификации (Information technology — Metadata registries (MDR) — Part 5: Naming and identification principles)

[15] ИСО 13584-1:2001 (ISO 13584-1:2001) — Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 1. Обзор и основные принципы (Industrial automation systems and integration. Parts library. Part 1. Overview and fundamental principles)

[16] ИСО 13584-24:2003 (ISO 13584-24:2003)	Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 24. Логический ресурс. Логическая модель библиотеки поставщика (Industrial automation systems and integration — Parts library — Part 24: Logical resource: Logical model of supplier library)
[17] ИСО 13584-25:2004 (ISO 13584-25:2004)	Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 25. Логический ресурс: логическая модель библиотеки поставщика с агрегированными значениями и подробным содержанием (Industrial automation systems and integration — Parts library — Part 25: Logical resource: Logical model of supplier library with aggregate values and explicit content)
[18] ИСО 13584-26:2000 (ISO 13584-26:2000)	Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 26. Логический ресурс: идентификация поставщика информации (Industrial automation systems and integration — Parts library — Part 26: Logical resource: Information supplier identification)
[19] ИСО 13584-32:2010 (ISO 13584-32:2010)	Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 32. Ресурсы внедрения. Язык разметки онтологии продукции (Industrial automation systems and integration — Parts library — Part 32: Implementation resources: OntoML: Product ontology markup language)
[20] ИСО 13584-511:2006 (ISO 13584-511:2006)	Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 511. Механические системы и компоненты общего назначения. Справочный словарь по крепежу (Industrial automation systems and integration — Parts library — Part 511: Mechanical systems and components for general use — Reference dictionary for fasteners
[21] ИСО/ТС 23768-1:2011 (ISO/TS 23768-1:2011)	Роликовые подшипники. Библиотека деталей. Часть 1. Справочный словарь (Rolling bearings — Parts library — Part 1: Reference dictionary for rolling bearings)
[22] ИСО/ТС 29002-5:2009 (ISO/TS 29002-5:2009)	Промышленные автоматические системы и интеграция. Обмен характеристическими данными. Часть 5. Схема идентификации (Industrial automation systems and integration — Exchange of characteristic data — Part 5: Identification scheme)
[23] ИСО/ТС 29002-20:2010 (ISO/TS 29002-20:2010)	Промышленные автоматические системы и интеграция. Обмен характеристическими данными. Часть 20. Услуги по идентификации концептуального словаря (Industrial automation systems and integration –Exchange of characteristic data — Part 20: Concept dictionary resolution services)
[24] ИСО 80000(все части) (ISO 80000) (all parts)	Величины и единицы (Quantities and units)
[25] МЭК 80000(все части) (IEC 80000) (all parts)	Величины и единицы (Quantities and units)
[26] МЭК 60027(все части) (IEC 60027) (all parts)	Обозначения буквенные, применяемые в электротехнике (Letter symbols to be used in electrical technology)
[27] МЭК 60748(все части) (IEC 60748) (all parts)	Приборы полупроводниковые (Semiconductor devices)
[28] МЭК 61360-1:2009 (IEC 61360-1:2009)	Стандартные типы элементов данных с соответствующей схемой классификации для электрических компонентов. Часть 1. Определения, принципы и методы (Standard data elements types with associated classification scheme for electric items — Part 1: Definitions — Principles and methods)
[29] МЭК 61360-4-DB:2005 (IEC 61360-4-DB:2005)	Стандартные типы элементов данных с соответствующей схемой классификации для электрических компонентов. Часть 4. Набор признаков стандартных типов элементов данных, классов компонентов и терминов в соответствии с МЭК (Standard data element types with associated classification scheme for electric components — Part 4: IEC reference collection of standard data element types and component classes Free access to Database)
[30] МЭК 61360-5:2004 (IEC 61360-5:2004)	Стандартные типы элементов данных с соответствующей схемой классификации для электрических компонентов. Часть 5. Добавления к словарной схеме EXPRESS (Standard data element types with associated classification scheme for electric components — Part 5: Extensions to the EXPRESS dictionary schema)

УДК 658.52.011.56:006.354 ОКС 25.040.40 Т58

Ключевые слова: автоматизированные промышленные системы, интеграция, жизненный цикл систем, управление производством

Редактор *Т.С. Никифорова*
Технический редактор *А.И. Белов*
Корректоры *Е.М. Бородулина, Г.Н. Старкова*
Компьютерная верстка *А.С. Шаповаловой*

Сдано в набор 05.03.2014. Подписано в печать 21.03.2014. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 21,39. Уч.-изд. л. 17,14. Тираж 58 экз. Зак. 868.

Набрано в Издательском доме «Вебстер»
gost.e-ecolog.ru gost.e-ecolog.ru
Отпечатано в Калужской типографии стандартов, 248021 Калуга, ул. Московская, 256

gost.e-ecolog.ru