ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 13584-25— 2010

Системы промышленной автоматизации и интеграция

БИБЛИОТЕКА ДЕТАЛЕЙ

Часть 25

Логический ресурс. Логическая модель библиотеки поставщика с агрегированными значениями и подробным содержанием

ISO 13584-25:2004
Industrial automation systems and integration — Parts library — Part 25: Logical resource: Logical model of supplier library with aggregate values and explicit content

(IDT)

Издание официальное

Москва
Стандартинформ
2014

gost.e-ecolog.ru

Предисловие

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

Сведения о стандарте

1 ПОДГОТОВЛЕН Научно-техническим центром «ИНТЕК» 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 21 декабря 2010 г. № 875-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 13584-25:2004 «Системы промышленной автоматизации и интеграция. Библиотека деталей. Часть 25. Логический ресурс. Логическая модель библиотеки поставщика с агрегированными значениями и подробным содержанием» (ISO 13584-25:2004 «Industrial automation systems and integration — Parts library — Part 25: Logical resource: Logical model of supplier library with aggregate values and explicit content»).

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

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

© Стандартинформ, 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, определения и сокращения . . . 2
4 Структура . . . 5
4.1 Ресурсы общего вида . . . 5
4.2 Интегрированная модель библиотеки . . . 5
5 Основные понятия и предположения . . . 7
5.1 Агрегированное значение свойств . . . 7
5.2 Описание расширения класса . . . 7
6 Схема ISO13584_IEC61360_dictionary_aggregate_extension_schema . . . 8
6.1 ISO13584_IEC61360_dictionary_aggregate_extension_schema: введение . . . 8
6.2 ISO13584_IEC61360_dictionary_aggregate_extension_schema: определение объектов . . . 8
7 Схема ISO13584_aggregate_value_schema . . . 11
7.1 ISO13584_aggregate_value_schema: введение . . . 12
7.2 ISO13584_aggregate_value_schema: определение объектов . . . 12
7.3 ISO13584_aggregate_value_schema: определение правил . . . 14
7.4 ISO13584_aggregate_value_schema: определение функций . . . 14
8 Интегрированная информационная модель библиотеки LIIM 25 . . . 23
8.1 ISO13584_25_IEC61360_5_liim_schema: краткий листинг . . . 24
8.2 Требования к классам соответствия . . . 29
Приложение A (обязательное) Сокращенные наименования сущностей, приведенных в настоящем стандарте . . . 41
Приложение B (обязательное) Регистрация информационных объектов . . . 42
Приложение C (справочное) Расширенный листинг схемы ISO13584_25_IEC61360_5_library_implicit_schema . . . 43
Приложение D (обязательное) Требования к стандартным данным интегрированной информационной модели библиотеки LIIM 25 . . . 44
Приложение E (обязательное) Требования к методам реализации интегрированной информационной модели библиотеки LIIM 25 . . . 51
Приложение F (справочное) Диаграммы EXPRESS-G . . . 52
Приложение G (справочное) Примеры файлов интегрированной информационной модели библиотеки LIIM 25 с комментариями. Обмен общими моделями явного вида . . . 54
Приложение H (справочное) Примеры файлов интегрированной информационной модели библиотеки LIIM 25 с комментариями. Обмен функциональными моделями явного вида, соответствующими ИСО 13584-101 . . . 60
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации . . . 72

Введение

Комплекс международных стандартов ИСО 13584 разработан с целью представления данных библиотеки деталей и обмена этими данными с помощью компьютерной интерпретации для определения механизма, способного передавать данные библиотеки деталей независимо от любого приложения, использующего информационную систему этой библиотеки.

Данный комплекс стандартов состоит из частей, входящих в следующие серии: концептуальные описания, логические ресурсы, ресурсы реализации, методология описаний, проверка на соответствие, протокол обмена представлениями и стандартизованное содержание. Описание серий приведено в ИСО 13584-1. Настоящий стандарт входит в серию, относящуюся к логическим ресурсам.

В настоящем стандарте определены ресурсы общего вида, необходимые для моделирования библиотек поставщика, включающих в себя свойства, значения которых могут иметь агрегированную структуру, а возможное содержание описано в виде набора экземпляров. Кроме того, в настоящем стандарте приведены интегрированные информационные модели на языке EXPRESS, позволяющие выполнять обмен данными библиотек поставщика. Для понимания требований настоящего стандарта необходимо знание языка EXPRESS, определенного в ИСО 10303-11. Кроме того, необходимы базовые знания требований ИСО 13584-24 и ИСО 13584-42.

Ресурсы общего вида, определенные в настоящем стандарте, разработаны специалистами ИСО ТК 184/ПК4/РГ2 и МЭК ПК 3D с целью их включения в настоящий стандарт и в МЭК 61360-5. Оба комитета пришли к соглашению о том, что они не будут независимо друг от друга изменять и/или модифицировать представленные схемы на языке EXPRESS. Данное соглашение обеспечивает гармонизацию и возможность повторного использования результатов работ двух комитетов. Вследствие этого заявки на внесение поправок в схемы EXPRESS должны быть одобрены обоими комитетами.

Настоящий стандарт подготовлен Техническим комитетом ИСО/ТК 184 «Системы промышленной автоматизации и интеграция», Подкомитетом ПК 4 «Производственные данные».

Комплекс стандартов ИСО 13584 состоит из следующих частей, имеющих общее название «Системы промышленной автоматизации и интеграция. Библиотека деталей»:

- Часть 1. Обзор и основные принципы;
- Часть 20. Логический ресурс. Логическая модель представлений;
- Часть 24. Логический ресурс. Логическая модель библиотеки поставщика;
- Часть 25. Логический ресурс. Логическая модель библиотеки поставщика с агрегированными значениями и подробным содержанием;
- Часть 26. Логический ресурс. Идентификация поставщика информации;
- Часть 31. Средства реализации. Интерфейс геометрического программирования;
- Часть 42. Методология описания. Методология структурирования групп деталей;
- Часть 101. Протокол обмена геометрическими представлениями по параметрической программе;
- Часть 102. Протокол обмена представлениями по спецификации соответствия ИСО 10303.

Описание структуры комплекса стандартов ИСО 13584 приведено в ИСО 13584-1. Нумерация частей ИСО 13584 отражает структуру комплекса:

- части 10—19 устанавливают требования к концептуальным описаниям;
- части 20—29 устанавливают требования к логическим ресурсам;
- части 30—39 устанавливают требования к ресурсам реализации;
- части 40—49 устанавливают методологию описания;
- части 100—199 устанавливают требования к протоколам обмена представлениями.

При публикации дополнительных частей ИСО 13584 должен использоваться этот же принцип их нумерации.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы промышленной автоматизации и интеграция

БИБЛИОТЕКА ДЕТАЛЕЙ

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

Дата введения — 2011—09—01

1 Область применения

Настоящей стандарт распространяется на конструкции ресурсов на языке EXPRESS общего вида, предназначенные для описания агрегированных типов данных и значений, которые входят в библиотеки поставщика. Кроме этого в стандарте приведена интегрированная информационная модель на языке EXPRESS для представления библиотек поставщика с целью обмена данными. Эта интегрированная модель объединяет в единую схему упомянутые выше конструкции ресурсов и другие конструкции ресурсов EXPRESS, описанные в других стандартах комплексов ИСО 13584 и ИСО 10303. Библиотеки поставщика состоят из определений и представлений семейств деталей, а также определений новых категорий представлений. Библиотеки поставщика должны включать в себя только элементы словарей с агрегированными типами данных либо без них и содержать дополнительные спецификации наборов допустимых экземпляров.

При использовании вместе с протоколами обмена видами данная интегрированная информационная модель предоставляет возможность обмена одной или несколькими категориями представлений деталей, определенными в библиотеке деталей.

Настоящий стандарт распространяется на:

- конструкции ресурсов общего вида для представления агрегированных типов данных. Типы и значения агрегированных данных моделируются в соответствии с определением агрегированных типов данных языка EXPRESS согласно ИСО 10303-11;
- конструкции ресурсов общего вида для представления агрегированных значений;
- конструкции ресурсов общего вида для представления деталей, которые могут содержать неограниченное число сборочных компонентов;
- интегрированную информационную модель библиотеки, предназначенную для моделирования и обмена библиотеками поставщика и содержащую свойства, значения которых могут иметь агрегированную структуру, а описания возможных расширений классов этих свойств приведены в виде наборов экземпляров.

Настоящий стандарт не распространяется на:

- представление выражений и переменных;
- описание набора допустимых экземпляров класса путем указания ограничений;
- спецификацию системы программного обеспечения, обеспечивающей управление библиотеками поставщика, представление которых соответствует информационным моделям, определенным в настоящем стандарте.

Издание официальное

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:

МЭК 61360-2:1998 Стандартные типы элементов данных с соответствующей схемой классификации для электрических компонент. Часть 2. Схема словаря EXPRESS (IEC 61360-2:1998, Standard data element types with associated classification scheme for electric components — Part 2: EXPRESS dictionary schema)

ИСО/МЭК 8824-1 Информационные технологии. Обозначения абстрактного синтаксиса 1 (ASN.1). Часть 1. Спецификация базовых обозначений (ISO/IEC 8824-1, Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation)

ИСО 8859-1 Информационные технологии. Наборы графических символов, кодируемых одним 8-битовым байтом. Часть 1. Латинский алфавит номер 1 (ISO 8859-1, Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1)

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

ИСО 10303-21:2002 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 21. Методы реализации. Кодирование открытого текста структуры обмена (ISO 10303-21:2002, Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure)

ИСО 10303-41:2000 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 41. Интегрированный ресурс общего вида. Основы описания и технического сопровождения продукции (ISO 10303-41:2000, Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support)

ИСО 10303-42:2003 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 42. Интегрированный ресурс общего вида. Геометрическое и топологическое представление (ISO 10303-42:2003, Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation)

ИСО 10303-43:2000 Системы промышленной автоматизации и интеграция. Представление данных о продукции и обмен данными. Часть 43. Интегрированный ресурс общего вида. Структуры представления (ISO 10303-43:2000, Industrial automation systems: and integration — Product data representation and exchange — Part 43: Integrated generic resource: Representation structures)

ИСО 13584-24:2003 Системы промышленной автоматизации и интеграция. Библиотека деталей. Часть 24. Логический ресурс. Логическая модель библиотеки поставщика (ISO 13584-24:2003, Industrial automation systems and integration — Parts library — Part 24: Logical resource: Logical model of supplier library)

ИСО 13584-42:1998 Системы промышленной автоматизации и интеграция. Библиотека деталей. Часть 42. Методология описания. Методология структурирования семейств деталей (ISO 13584-42:1998, Industrial automation systems and integration — Parts library — Part 42: Description methodology: Methodology for structuring part families)

IAB RFC 2068 Протокол передачи гипертекста HTTP/1.1 (HTTP-1.1), Стандартный протокол, предлагаемый сообществом пользователей для архитектуры Интернет (IAB RFC 2068, Hypertext transfer protocol HTTP/1.1 (HTTP-1.1), Internet architecture board proposed standard protocol)

3 Термины, определения и сокращения

В настоящем стандарте использованы термины по ИСО 10303-1, ИСО 10303-11, ИСО 13584-1, ИСО 13584-24, ИСО 13584-42.

3.1 **применимое свойство** (applicable property): Свойство конкретного семейства деталей, которое должно использоваться для всех деталей, принадлежащих этому семейству.

[ИСО 13584-24:2003]

Пример — Для общего семейства винтовых деталей применимым свойством является диаметр резьбы. Эту характеристику следует применять для любого винта.

3.2 **базовая семантическая единица** (basic semantic unit; BSU): Элемент, предназначенный для абсолютной и универсальной идентификации определенных объектов в прикладной области, например для классов и типов элементов данных.

[ИСО 13584-42:1998]

3.3 **расширение класса** (class extension): Совокупность экземпляров, соответствующих спецификации, определяемой классом.

[ИСО 13584-24:2003]

3.4 **общая схема словаря** (common dictionary schema): Информационная модель словаря, представленная с использованием языка моделирования EXPRESS в результате совместных разработок специалистов ИСО ТК 184/ПК4/РГ2 и МЭК ПК 3D.

[ИСО 13584-42:1998]

Примечание — Общая схема словаря имеет формальное наименование ISO13584_IEC61360_dictionary_schema и определена согласно МЭК 61360-2. Данная схема приведена в приложении D ИСО 13584-42.

3.5 **класс соответствия** (conformance class): Подмножество требований стандарта, для которых установлено соответствие.

[ИСО 13584-24:2003]

3.6 **требование соответствия** (conformance requirement): Точное описание характеристики, которая должна присутствовать в реализации для соответствия этому требованию.

[ИСО 10303-1:1994]

3.7 **элемент словаря** (dictionary element): Набор атрибутов, составляющих словарное описание определенных объектов прикладной области.

Примечание — К набору атрибутов можно отнести классы и типы элементов данных.

[ИСО 13584-42:1998]

3.8 **тип элемента данных** (data element type; DET): Единица данных, для которой задана идентификация, описание и представление значений.

[ИСО 13584-42:1998]

3.9 **тип данных** (data type): Область значений.

[ИСО 10303-11:1994]

3.10 **семейство деталей** (family of parts): Простое или общее семейство деталей.

[ИСО 13584-24:2003]

3.11 **функциональная модель детали** (functional model of a part): Библиотечные данные, предназначенные для описания конкретной категории представления детали в интегрированной библиотеке.

[ИСО 13584-1:2001]

3.12 **функциональный вид представления детали** (functional view of a part): Данные, предназначенные для описания конкретной категории представления детали в наборе данных об изделии.

[ИСО 13584-1:2001]

Примечание — Структура функционального вида не зависит от детали, которую данный вид представляет.

3.13 **общая модель детали** (general model of a part): Библиотечные данные, содержащие определение и позволяющие идентифицировать деталь в интегрированной библиотеке.

[ИСО 13584-1:2001]

3.14 **общее семейство деталей** (generic family of parts): Объединение деталей в простые или общие группы с целью классификации или представления информации, общей для этих групп.

[ИСО 13584-24:2003]

3.15 **файл поставки библиотеки** (library delivery file): Совокупность экземпляров сущностей языка EXPRESS, которые соответствуют интегрированной информационной модели библиотеки и представлены с помощью одного из методов, указанных в ИСО 10303.

[ИСО 13584-24:2003]

Примечание — Файл поставки библиотеки устанавливает структуру и содержимое поставки библиотеки. Также в нем могут быть ссылки на внешние библиотечные файлы.

3.16 **библиотечная деталь** (library part): Деталь с соответствующим набором данных, представленная в библиотеке.

[ИСО 13584-1:2001]

3.17 **данные библиотечной детали** (library part data): Данные, представляющие деталь в библиотеке.

[ИСО 13584-1:2001]

3.18 **содержание библиотечного обмена** (library exchange context): Файл поставки библиотеки и один или несколько внешних библиотечных файлов, в совокупности представляющие библиотеку поставщика.

[ИСО 13584-24:2003]

3.19 **внешний библиотечный файл** (library external file): Файл, дополняющий определение библиотеки поставщика, на который есть ссылка в файле поставки библиотеки.

[ИСО 13584-24:2003]

Примечание — Структура и формат внешнего библиотечного файла определены в файле поставки библиотеки, в котором имеется ссылка на этот файл.

3.20 **интегрированная информационная модель библиотеки** (library integrated information model; LIIM): Схема на языке EXPRESS с интегрированными конструкциями ресурсов из разных схем на языке EXPRESS для представления библиотек поставщика с целью обмена, ассоциированная с требованиями соответствия.

[ИСО 13584-24:2003]

3.21 **библиотечная спецификация класса** (library specification of a class): Точное представление расширения класса в библиотеке поставщика.

[ИСО 13584-24:2003]

Примечание 1 — В настоящем стандарте каждый класс определен с помощью элементов данных. Библиотечная спецификация есть только у тех классов, для которых поставщик может представить возможные экземпляры.

Примечание 2 — В настоящем стандарте библиотечная спецификация класса состоит из набора, содержащего все возможные экземпляры.

3.22 **деталь** (part): Материал или функциональный элемент, предназначенный для включения в разные изделия в качестве компонента.

[ИСО 13584-1:2001]

3.23 **свойство** (property): Информация, которая может быть представлена типом элемента данных.

[ИСО 13584-42:1998]

3.24 **категория представления** (representation category): Абстрактное понятие, используемое для разделения пользовательских требований, имеющих отношение к представлению детали.

[ИСО 13584-1:2001]

Примечание — В модели, определенной в комплексе стандартов ИСО 13584, такое разделение формально выражено в терминах логических имен видов и управляющих переменных видов.

3.25 **конструкция ресурса** (resource construct): Совокупность объектов, типов, функций, правил и ссылок на языке EXPRESS, определяющая допустимое описание данных.

[ИСО 13584-24:2003]

3.26 **простое семейство деталей** (simple family of parts): Набор деталей, каждая из которых может быть описана одной и той же группой свойств.

[ИСО 13584-24:2003]

3.27 **библиотека поставщика** (supplier library): Набор данных и программ, описывающих в стандартном формате по ИСО 13584 набор деталей и/или набор представлений деталей.

[ИСО 13584-1:2001]

3.28 **библиотека пользователя** (user library): Информация, полученная в результате интеграции одной или нескольких библиотек поставщика в систему управлений библиотекой с последующей их адаптацией пользователем.

[ИСО 13584-1:2001]

3.29 **протокол обмена видами** (view exchange protocol; VEP): Описание использования конструкций ресурсов и представления интерфейсов передачи, соответствующих требованиям к обмену информацией о деталях одной категории представления.

[ИСО 13584-24:2003]

3.30 **видимое свойство** (visible property): Свойство семейства деталей, применимое либо не применимое для разных деталей этого семейства.

[ИСО 13584-24:2003]

Пример — Видимым свойством общего семейства винтов является длина участка без резьбы. Это свойство определено для любого винта, но только у винтов с участком без резьбы имеется значение данного свойства.

Примечание — Код класса, в котором определено свойство, является видимой частью идентификации для типа элемента данных, представляющего это свойство.

4 Структура

Настоящий стандарт состоит из двух основных частей:

- ресурсов общего вида, устанавливающих требования к конструкции ресурсов для представления типов и значений агрегированных данных, определенных на языке EXPRESS;
- интегрированной информационной модели библиотеки LIIM, объединяющей конструкции ресурсов и конструкции ресурсов общего вида, приведенные в других стандартах комплекса ИСО 13584 и ИСО 10303 в рамках единой схемы представления библиотек поставщика, в которую входят агрегированные типы данных и агрегированные значения, а также явные описания расширений классов в виде наборов экземпляров.

4.1 Ресурсы общего вида

Ресурсы общего вида представляют с помощью следующих схем на языке EXPRESS:

- ISO13584_IEC61360_dictionary_aggregate_extension_schema;
- ISO13584_aggregate_value_schema.

Эти схемы определяют общие конструкции ресурсов, применяющиеся не только в рамках комплекса стандартов ИСО 13584, но и во всех приложениях, где используется словарь данных, соответствующий стандартам серии МЭК 61360.

4.1.1 Схема ISO13584_IEC61360_dictionary_aggregate_extension_schema

ISO13584_IEC61360_dictionary_aggregate_extension_schema определяет конструкции ресурсов, необходимые для описания типов данных, соответствующих агрегированным типам данных, включая типы данных, определенные на языке EXPRESS. В данной схеме определены ресурсы для описания типов данных массивов, мультимножеств, списков, множеств и наборов подмножеств. Эти типы данных являются расширениями типов данных, определенных в **ISO13584_IEC61360_dictionary_schema**, установленной в МЭК 61360-2, содержание которой приведено в ИСО 13584-42.

4.1.2 Схема ISO13584_aggregate_value_schema

ISO13584_aggregate_value_schema определяет конструкции ресурсов, необходимые для описания значений типов данных, которые соответствуют агрегированным типам данных, определенным на языке EXPRESS. В данной схеме определены ресурсы для описания значений массивов, мультимножеств, списков и структурированных наборов. Эти значения данных являются расширениями значений, определенных в ISO13584_instance_resource_schema, установленной в ИСО 13584-24.

4.2 Интегрированная модель библиотеки

Интегрированная информационная модель библиотеки LIIM 25 объединяет конструкции ресурсов общего вида, определенные в настоящем стандарте, в других стандартах комплекса ИСО 13584, а также в ИСО 10303, в единую схему, используемую для представления библиотек поставщика с целью обмена данными. Модель библиотеки LIIM 25 позволяет поставщику библиотечных данных и конечному пользователю библиотеки обмениваться словарями или библиотеками семи типов, а также наборами экземпляров библиотек, не имеющих библиотечной структуры. Для обмена могут использоваться:

- словари класса соответствия 1, в которых определены иерархии классов таких элементов, как детали, материалы и т. д., а в свойствах агрегированной структуры применяются только конструкции ресурсов на языке EXPRESS, определенные в общей схеме словаря ИСО/МЭК или в **ISO13584_IEC61360_dictionary_aggregate_extension_schema**, соответствующей требованиям настоящего стандарта;
- словари класса соответствия 2, в которых определены иерархии классов таких элементов, как детали, материалы, конструктивные особенности и т. д., и используется расширение общей схемы словаря ИСО 13584-24 без описания представлений элементов и категорий элементов, а также свойств агрегированной структуры;

- словари класса соответствия 3, в которых определены иерархии классов элементов, представлений элементов и их категорий и используются свойства агрегированной структуры;

- словари класса соответствия 4, назначение которых аналогично назначению словарей класса соответствия 3, но имеющие не более двух уровней вложения свойств агрегированной структуры;

- библиотеки класса соответствия 5, определяющие набор экземпляров, входящих в некоторые иерархии классов элементов, без представлений классов элементов и категорий элементов, а также свойств агрегированной структуры;

- библиотеки класса соответствия 6, определяющие набор экземпляров, входящих в некоторые иерархии классов элементов и представлений элементов и использующие свойства агрегированной структуры;

- библиотеки класса соответствия 7, назначение которых аналогично назначению библиотек класса соответствия 6, но имеющие не более двух уровней вложения свойств агрегированной структуры;

- набор экземпляров элементов и представлений элементов класса соответствия 10 без словарных определений и без библиотечной структуры;

- набор экземпляров элементов и представлений элементов класса соответствия 11 без библиотечной структуры, но с возможными словарными определениями.

Каждый элемент обмена содержанием, приведенный выше, соответствует конкретному классу соответствия LIIM 25. Каждый класс соответствия устанавливает требования соответствия для реализаций, претендующих на соответствие данному классу. В классах соответствия 6 и 7 поддерживаются все объекты, типы и соответствующие конструкции, которые являются частью LIIM 25. В других классах соответствия обеспечивается только поддержка подмножества этого набора конструкций ресурсов. В настоящем стандарте каждое подмножество, определяющее класс соответствия, установлено путем указания списка объектов. Реализация, претендующая на соответствие какому-либо классу соответствия, должна поддерживать все сущности, указанные для данного класса соответствия и связанных конструкций.

4.2.1 Класс соответствия 1: минимальные словари

Класс соответствия 1 поддерживает информационные требования обмена определениями иерархий классов таких элементов, как детали и материалы. Для данного класса соответствия возможен обмен всеми элементами схемы словаря ИСО/МЭК и классами элементов, свойства которых имеют агрегированные значения. Класс соответствия 1 ассоциируется с методами реализации файла поставки библиотеки. Требования к классу соответствия 1 модели LIIM 25 приведены в 8.2.1.

4.2.2 Класс соответствия 2: словари классов элементов

Класс соответствия 2 поддерживает информационные требования обмена определениями иерархий классов таких элементов, как детали, материалы и конструктивные особенности, свойства которых не имеют агрегированных значений. Класс соответствия 2 ассоциируется с набором стандартных данных, определяющим форматы всех внешних библиотечных файлов, на которые могут быть ссылки из файла поставки библиотеки класса соответствия 2 модели LIIM 25, а также с методами реализации файла поставки библиотеки. Требования к классу соответствия 2 модели LIIM 25 приведены в 8.2.2.

4.2.3 Класс соответствия 3: полные словари

Класс соответствия 3 поддерживает информационные требования обмена определениями иерархий классов таких элементов, как детали, материалы и конструктивные особенности, дополненными определениями представлений и категорий представлений этих классов элементов. Свойства всех классов имеют агрегированные значения. Класс соответствия 3 ассоциируется с набором стандартных данных, определяющим форматы всех внешних библиотечных файлов, на которые могут быть ссылки из файла поставки библиотеки, соответствующего классу соответствия 3 модели LIIM 25, а также с методами реализации файла поставки библиотеки. Требования к классу соответствия 3 модели LIIM 25 приведены в 8.2.3.

4.2.4 Класс соответствия 4: полные словари с агрегированными значениями ограниченного уровня вложения

Класс соответствия 4 поддерживает информационные требования класса соответствия 3, за исключением того, что уровень вложения агрегированных значений класса соответствия 4 должен быть не более 2. Требования к классу соответствия 4 модели LIIM 25 приведены в. 8.2.4.

4.2.5 Класс соответствия 5: библиотеки классов элементов

Класс соответствия 5 поддерживает информационные требования обмена определениями и экземплярами иерархий классов таких элементов, как детали, материалы и конструктивные особенности, свойства которых могут не иметь агрегированных значений. Расширения классов могут быть определены только с помощью наборов экземпляров, без использования каких-либо условий или выражений. Класс соответствия 5 ассоциируется с набором стандартных данных, определяющим форматы всех внешних библиотечных файлов, на которые могут быть ссылки из файла поставки библиотеки, соответствующего классу

соответствия 5 модели LIIM 25, а также с методами реализации файла поставки библиотеки. Требования к классу соответствия 5 модели LIIM 25 приведены в 8.2.5.

4.2.6 Класс соответствия 6: полные библиотеки

Класс соответствия 6 поддерживает информационные требования обмена определениями и экземплярами иерархий классов таких элементов, как детали, материалы и конструктивные особенности, дополненными определениями и экземплярами представлений и категорий представлений этих классов элементов. Свойства всех классов могут иметь агрегированные значения. Расширения классов могут быть определены только с помощью наборов экземпляров, без использования каких-либо условий или выражений. Класс соответствия 6 ассоциируется с набором стандартных данных, определяющим форматы всех внешних библиотечных файлов, на которые могут быть ссылки из файла поставки библиотеки, соответствующего классу соответствия 6 модели LIIM 25, а также с методами реализации файла поставки библиотеки. Требования к классу соответствия 6 модели LIIM 25 приведены в 8.2.6.

4.2.7 Класс соответствия 7: полные библиотеки с агрегированными значениями ограниченного уровня вложения

Класс соответствия 7 поддерживает информационные требования класса соответствия 6, за исключением того, что уровень вложения агрегированных значений данного класса соответствия должен быть не более 2. Требования к классу соответствия 7 модели LIIM 25 приведены в 8.2.7.

4.2.8 Класс соответствия 10: экземпляры библиотек

Класс соответствия 10 поддерживает информационные требования обмена одним или несколькими экземплярами таких элементов, как детали, материалы и конструктивные особенности, или обмена представлениями элементов, не содержащими словарных определений. Данный класс соответствия предназначен для предоставления ресурсов, необходимых для поставки набора экземпляров, выбранных пользователем из библиотеки и, возможно, дополненных представлениями элементов. Класс соответствия 10 ассоциируется с набором стандартных данных, определяющим форматы всех внешних библиотечных файлов, на которые могут быть ссылки из файла поставки библиотеки, соответствующего классу соответствия 10 модели LIIM 25, а также с методами реализации файла поставки библиотеки. Требования к классу соответствия 10 модели LIIM 25 приведены в 8.2.8.

4.2.9 Класс соответствия 11: экземпляры библиотек с соответствующими словарными определениями

Класс соответствия 11 поддерживает информационные требования обмена одним или несколькими экземплярами таких элементов, как детали, материалы и конструктивные особенности, дополненными одним или несколькими экземплярами представлений этих элементов без использования библиотечной структуры, но с возможным использованием словарных определений. Данный класс соответствия предназначен для предоставления ресурсов, необходимых для поставки набора экземпляров, выбранных пользователем из библиотеки и, возможно, дополненных представлениями элементов, если у пользователя отсутствуют словари, используемые в библиотеке. Класс соответствия 11 ассоциируется с набором стандартных данных, определяющим форматы всех внешних библиотечных файлов, на которые могут быть ссылки из файла поставки библиотеки, соответствующего классу соответствия 11 модели LIIM 25, а также с методами реализации файла поставки библиотеки. Требования к классу соответствия 11 модели LIIM 25 приведены в 8.2.9.

5 Основные понятия и предположения

В настоящем стандарте применены следующие понятия и предположения.

5.1 Агрегированное значение свойств

В схемах **ISO13584_IEC61360_dictionary_aggregate_extension_schema** и **ISO13584_aggregate_value_schema** определены конструкции ресурсов общего вида для представления агрегированных типов данных, а также конструкции ресурсов общего вида для представления агрегированных значений, в которые входят типы данных и значения, определенные на языке EXPRESS по ИСО 10303-11. Указанные ресурсы позволяют устанавливать агрегированные типы данных и значения любым свойствам любых экземпляров каждого класса, описанного в библиотеке.

5.2 Описание расширения класса

Интегрированная информационная модель библиотеки, определенная в настоящем стандарте, поддерживает моделирование и обмен для библиотек поставщика, содержащих свойства, значения которых

могут иметь агрегированную структуру, а возможные расширения классов представлены наборами экземпляров. Данная интегрированная информационная модель библиотеки обеспечивает точное представление библиотек в виде набора экземпляров, ассоциируемых со значениями свойств.

6 Схема ISO13584_IEC61360_dictionary_aggregate_extension_schema

В данном разделе установлены требования к **ISO13584_IEC61360_dictionary_aggregate_extension_schema**. Для ознакомления с **ISO13584_IEC61360_dictionary_aggregate_extension_schema** и определения необходимых внешних ссылок используют следующее объявление на языке EXPRESS:

Спецификация EXPRESS:

```
*)
SCHEMA ISO13584_IEC61360_dictionary_aggregate_extension_schema;
REFERENCE FROM ISO13584_IEC61360_dictionary_schema (data_type, entity_instance_type);
(*
```

Примечание — Схема, описанная выше, определена в **ISO13584_IEC61360_dictionary_schema** МЭК 61360-2, а также приведена в приложении D ИСО 13584-42.

6.1 ISO13584_IEC61360_dictionary_aggregate_extension_schema: введение

Схема **ISO13584_IEC61360_dictionary_aggregate_extension_schema** предоставляет информационную модель, используемую для расширения общей словарной схемы ИСО/МЭК, которая позволяет использовать списки, множества, массивы и наборы подмножеств простых или сложных типов данных.

Данное расширение реализуют в два этапа.

- сначала вводят объект **entity_instance_type_for_aggregate**, предоставляющий средства ссылки на объекты EXPRESS, определяющие агрегированные типы данных. Этот объект является подтипом объекта **entity_instance_type**.

Примечание — Требования к **entity_instance_type** установлены в МЭК 61630-2 и ИСО 13584-42:1988;

- затем объекты, устанавливающие агрегированные типы данных, конструируют с помощью объекта **aggregate_type** и его конкретизаций.

6.2 ISO13584_IEC61360_dictionary_aggregate_extension_schema: определение объектов

Следующие определения типов объектов описывают ресурсы, необходимые для представления агрегированных типов данных.

6.2.1 Объект aggregate_entity_instance_type

Объект **entity_instance_type_for_aggregate** предоставляет возможность привести ссылку на определения типов данных, которые могут быть выражены в виде списков, множеств, мультимножеств или массивов простых или сложных значений. Данный объект определяется ссылкой на тип **aggregate_type**, определенный в данной схеме.

Спецификация EXPRESS:

```
*)
ENTITY entity_instance_type_for_aggregate
SUBTYPE OF(entity_instance_type);
        type_structure: aggregate_type;
WHERE
        WR1:SELF\entity_instance_type.type_name =
        ['ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA'
        + '.AGGREGATE_TYPE'];
END_ENTITY;
(*
```

Определение атрибутов:

type_structure: объект типа **aggregate_type**, ссылки на который и использование которого обеспечиваются объектом **entity_instance_type**.

Формальные положения:

WR1: атрибут **type_name** объекта **entity_instance_type** должен содержать строку:
'ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA.AGGREGATE_TYPE'.

6.2.2 Объект aggregate_type

Объект **aggregate_type** обеспечивает определение типов данных, которые могут быть выражены в виде списков, множеств, мультимножеств или массивов простых или сложных значений.

Спецификация EXPRESS:

```
*)
ENTITY aggregate_type
ABSTRACT SUPERTYPE OF(ONEOF(
                list_type,
                set_type,
                bag_type,
                array_type, set_with_subset_constraint_type ));
        bound_1: OPTIONAL INTEGER;
        bound_2: OPTIONAL INTEGER;
        value_type: data_type;
WHERE
        WR1: bound_1 <= bound_2;
END_ENTITY;
(*
```

Определение атрибутов:

value_type: тип значения (простого или сложного), используемого для каждого элемента агрегированного типа.

bound_1: необязательное целое число, устанавливающее нижнюю границу определяемого агрегированного типа.

bound_2: необязательное целое число, устанавливающее верхнюю границу определяемого агрегированного типа.

Формальные положения:

WR1: **bound_1** не может быть больше **bound_2**.

6.2.3 Объект list_type

Объект **list_type** обеспечивает определение типов данных, выраженных в виде упорядоченного списка значений, в котором допустимо или недопустимо дублирование элементов.

Спецификация EXPRESS:

```
*)
ENTITY list_type
SUBTYPE OF(aggregate_type);
        uniqueness: BOOLEAN;
WHERE
        WR1: EXISTS(bound_1) OR NOT(EXISTS(bound_2));
        WR2: NOT(EXISTS(bound_1)) OR (bound_1 >= 0);
END_ENTITY;
(*
```

Определение атрибутов:

uniqueness: флаг, предназначенный для указания того, что все элементы списка должны быть уникальными (значение true), либо они могут дублироваться (значение false).

Формальные положения:

WR1: если в определяемом списке присутствует необязательный атрибут верхней границы **bound_2**, в нем также есть необязательный атрибут нижней границы **bound_1**.

WR2: если в определяемом списке присутствует необязательный атрибут нижней границы **bound_1**, то значение этого атрибута не должно быть неотрицательным.

6.2.4 Объект set_type

Объект **set_type** обеспечивает определение типов данных, выраженных в виде неупорядоченного набора значений, в котором не допускается наличие повторяющихся элементов.

Спецификация EXPRESS:

```
*)
ENTITY set_type
SUBTYPE OF(aggregate_type);
```

```
WHERE
        WR1: EXISTS (bound_1) OR NOT (EXISTS(bound_2));
        WR2: NOT (EXISTS(bound_1)) OR (bound_1 >= 0);
END_ENTITY;
(*
```

Формальные положения:

WR1: если в определяемом наборе присутствует необязательный атрибут верхней границы **bound_2**, в нем также есть необязательный атрибут нижней границы **bound_1**.

WR2: если в определяемом наборе присутствует необязательный атрибут нижней границы **bound_1**, то значение этого атрибута не должно быть отрицательным.

6.2.5 Объект bag_type

Объект **bag_type** обеспечивает определение типов данных, выраженных в виде неупорядоченного набора значений, в котором могут присутствовать повторяющиеся элементы.

Спецификация EXPRESS:

```
*)
ENTITY bag_type
SUBTYPE OF(aggregate_type);
WHERE
        WR1: EXISTS(bound_1) OR NOT(EXISTS(bound_2));
        WR2: NOT(EXISTS(bound_1)) OR (bound_1 >= 0);
END_ENTITY;
(*
```

Формальные положения:

WR1: если в определяемом наборе присутствует необязательный атрибут верхней границы **bound_2**, в нем есть необязательный атрибут нижней границы **bound_1**.

WR2: если в определяемом наборе присутствует необязательный атрибут нижней границы **bound_1**, то значение этого атрибута должно быть неотрицательным.

6.2.6 Объект array_type

Объект **array_type** обеспечивает определение типов данных, выраженных в виде массива значений. Областью значений этого типа данных является индексированный набор аналогичных элементов фиксированного размера. Нижняя и верхняя границы массива, выраженные целыми числами, определяют диапазон значений индекса и, как следствие, размер массива. При определении типа данных массива может быть дополнительно указано, что значение массива не должно содержать повторяющихся элементов.

Спецификация EXPRESS:

```
*)
ENTITY array_type
SUBTYPE OF (aggregate_type);
        SELF\aggregate_type.bound_1: INTEGER;
        SELF\aggregate_type.bound_2: INTEGER;
        uniqueness: BOOLEAN;
        are_optional: BOOLEAN;
END_ENTITY;
(*
```

Определение атрибутов:

bound_1: целое значение, устанавливающее нижний индекс определяемого агрегированного типа.

bound_2: целое значение, устанавливающее верхний индекс определяемого агрегированного типа.

uniqueness: указывает на то, что все элементы массива должны быть уникальными (значение true), либо они могут дублироваться (значение false).

are_optional: указывает на то, что в массив должны входить все элементы (значение false), либо на то, что некоторые элементы массива могут отсутствовать (значение true).

6.2.7 Объект set_with_subset_constraint_type

Объект **set_with_subset_constraint_type** обеспечивает определение типов данных, выраженных в виде множества значений, в котором могут существовать подмножества. Размеры допустимых подмножеств определяются минимальным и максимальным значениями. Если эти значения отсутствуют, допускается любое подмножество.

Примечание — Настоящий стандарт не распространяется на контекст, в котором могут быть выделены подмножества.

Спецификация EXPRESS:

```
*)
ENTITY set_with_subset_constraint_type
SUBTYPE OF (aggregate_type);
        cardinal_min: OPTIONAL INTEGER;
        cardinal_max: OPTIONAL INTEGER;
WHERE
        WR1: cardinal_min <= cardinal_max ;
        WR2: NOT EXISTS (bound_2) OR NOT EXISTS (cardinal_max)
                OR (cardinal_max <= bound_2);
        WR3: NOT EXISTS (bound_1) OR NOT EXISTS (cardinal_min)
                OR (cardinal_min <= bound_1);
END_ENTITY;
(*
```

Определение атрибутов:

cardinal_min: минимальный размер подмножеств, которые могут быть выделены.

cardinal_max: максимальный размер подмножеств, которые могут быть выделены.

Формальные положения:

WR1: минимальный размер выделяемых подмножеств **cardinal_min** не может быть больше максимального размера выделяемых подмножеств cardinal_max.

WR2: максимальный размер выделяемых подмножеств **cardinal_min** не может быть больше максимального размера самого множества.

WR3: минимальный размер выделяемых подмножеств не может быть меньше минимального размера самого множества.

```
*)
END_SCHEMA;
-- ISO13584_IEC61360_dictionary_aggregate_extension_schema
(*
```

7 Схема ISO13584_aggregate_value_schema

В данном разделе установлены требования к **ISO13584_aggregate_value_schema**. Для определения **ISO13584_aggregate_value_schema** и указания необходимых внешних ссылок используют следующее объявление на языке EXPRESS:

Спецификация EXPRESS:

```
*)
SCHEMA ISO13584_aggregate_value_schema;
REFERENCE FROM ISO13584_IEC61360_dictionary_schema (class_instance_type, data_type,
        data_type_element, level_type, named_type, property_BSU, property_DET);
REFERENCE FROM ISO13584_IEC61360_dictionary_aggregate_extension_schema (
        entity_instance_type_for_aggregate, list_type, set_type, bag_type, array_type,
        set_with_subset_constraint_type);
REFERENCE FROM ISO13584_extended_dictionary_schema (data_type_type_name,
        data_type_typeof);
REFERENCE FROM ISO13584_instance_resource_schema (compatible_class_and_class,
        compatible_level_type_and_instance, compatible_type_and_value, dic_class_instance,
        null_or_primitive_value, primitive_value, property_or_data_type_BSU, property_value,
        uncontrolled_entity_instance_value);
(*
```

Примечание — Требования к вышеуказанным схемам установлены в следующих документах:

- МЭК 61360-2 — **ISO13584_IEC61360_dictionary_schema** (данная схема также приведена в приложении D ИСО 13584-42);
- настоящий стандарт — **ISO13584_IEC61360_dictionary_aggregate_extension_schema**;
- ИСО 13584-24 — **ISO13584_extended_dictionary_schema**;
- ИСО 13584-24 — **ISO13584_instance_resource_schema**.

7.1 ISO13584_aggregate_value_schema: введение

ISO13584_aggregate_value_schema представляет информационную модель расширения **ISO_13584_instance_resource_schema**, которая обеспечивает распространение значений экземпляров на экземпляры агрегированных типов данных, таких как списки, множества, мультимножества, массивы или наборы подмножеств простых или сложных типов данных в соответствии с общей библиотечной схемой ИСО/МЭК **ISO13584_IEC61360_dictionary_aggregate_extension_schema**.

7.2 ISO13584_aggregate_value_schema: определение объектов

Приведенные в настоящем подразделе определения типов объектов описывают ресурсы, необходимые для представления агрегированных значений с помощью языка EXPRESS.

7.2.1 Объект aggregate_entity_instance_value

Объект **aggregate_entity_instance_value** обеспечивает ссылки на значения данных, выраженные в виде совокупностей примитивных значений **primitive_value**.

Данный объект определяют путем выделения подтипа **uncontrolled_entity_instance_value**, предоставляемого **ISO13584_instance_resource_schema** в соответствии с ИСО 13584-24.

Спецификация EXPRESS:

```
*)
ENTITY aggregate_entity_instance_value
SUBTYPE OF (uncontrolled_entity_instance_value);
        the_value: aggregate_value;
END_ENTITY;
(*
```

Определение атрибутов:

the_value: агрегированное значение, включающее в себя все значения описываемой сущности.

7.2.2 Объект aggregate_value

Объект **aggregate_value** обеспечивает определение значений данных, выраженных в виде совокупности примитивных значений **primitive_value**.

Спецификация EXPRESS:

```
*)
ENTITY aggregate_value
ABSTRACT SUPERTYPE OF(ONEOF(list_value, set_value, bag_value, array_value,
        set_with_subset_constraint_value ));
        values: LIST OF null_or_primitive_value;
END_ENTITY;
(*
```

Определение атрибутов:

values: список, включающий в себя все значения агрегированного типа описываемой сущности.

7.2.3 Объект list_value

Объект **list_value** обеспечивает определение значений данных, выраженных в виде списка, состоящего из примитивных значений **primitive_value**.

Спецификация EXPRESS:

```
*)
ENTITY list_value
SUBTYPE OF (aggregate_value);
        SELF\aggregate_value.values: LIST OF primitive_value;
END_ENTITY;
(*
```

Определение атрибутов:

values: список, включающий в себя все значения описываемой сущности.

7.2.4 Объект set_value

Объект **set_value** обеспечивает определение значений данных, выраженных в виде множества, состоящего из примитивных значений **primitive_value**.

Спецификация EXPRESS:

```
*)
ENTITY set_value
```

```
SUBTYPE OF (aggregate_value);
        SELF\aggregate_value.values: LIST OF primitive_value;
WHERE
        WR1: VALUE_UNIQUE(values);
END_ENTITY;
(*
```

Определение атрибутов:

values: список, включающий в себя все значения множества описываемой сущности.

Формальные положения:

WR1: значения в списке **values** должны быть уникальными; повторяющиеся значения недопустимы.

7.2.5 Объект bag_value

Объект **bag_value** обеспечивает определение значений данных, которые могут быть выражены в виде мультимножества, состоящего из примитивных значений **primitive_value**.

Спецификация EXPRESS:

```
*)
ENTITY bag_value
SUBTYPE OF (aggregate_value);
        SELF\aggregate_value.values: LIST OF primitive_value;
END_ENTITY;
(*
```

Определение атрибутов:

values: список, включающий в себя все значения мультимножества описываемой сущности.

7.2.6 Объект array_value

Объект **array_value** обеспечивает определение значений данных, выраженных в виде массива, состоящего из значений **null_or_primitive_value**.

Спецификация EXPRESS:

```
*)
ENTITY array_value
SUBTYPE OF (aggregate_value);
        bound_1: INTEGER;
        bound_2: INTEGER;
WHERE
        WR1: SIZEOF (SELF\aggregate_value.values) = SELF.bound_2 - SELF.bound_1 + 1;
END_ENTITY;
(*
```

Определение атрибутов:

bound_1: значение нижней границы массива.

bound_2: значение верхней границы массива.

Формальные положения:

WR1: размер атрибута списка **values** должен быть равен размеру, определенному атрибутами **bound_1** и **bound_2**.

7.2.7 Объект set_with_subset_constraint_value

Объект **set_with_subset_constraint_value** обеспечивает определение значений данных, выраженных в виде множества, состоящего из примитивных значений **primitive_value** и подмножеств. Размеры допустимых подмножеств определяют либо минимальным и максимальным значением, указанным для типа данных **set_with_subset_constraint_type**, либо атрибутами **min** и **max**. Если не установлены границы, допускается использовать любые подмножества.

Примечание 1 — Настоящий стандарт не распространяется на контекст, в котором могут быть выделены подмножества.

Примечание 2 — Если помимо **min** и **max** также указаны **cardinal_min** и **cardinal_max**, правило **allowed_aggregate_value** обеспечивает выполнение условия **cardinal_min <= min <= max <= cardinal_max**.

Спецификация EXPRESS:

```
*)
ENTITY set_with_subset_constraint_value
SUBTYPE OF (aggregate_value);
```

```
        SELF\aggregate_value.values: LIST OF primitive_value;
        min: OPTIONAL INTEGER;
        max: OPTIONAL INTEGER;
    WHERE
        WR1: NOT EXISTS (min) OR NOT EXISTS (max) OR (min <= max) ;
        WR2: VALUE_UNIQUE(values);
    END_ENTITY;
    (*
```

Определение атрибутов:

values: список, включающий в себя все значения множества, из которого могут быть выделены подмножества.

min: минимальный размер подмножеств, которые могут быть выделены.

max: максимальный размер подмножеств, которые могут быть выделены.

Формальные положения:

WR1: минимальный размер подмножеств **min** не может быть больше максимального размера подмножеств **max**.

WR2: все значения в списке **values** должны быть уникальными; повторяющиеся значения недопустимы.

7.3 ISO13584_aggregate_value_schema: определение правил

7.3.1 Правило allowed_aggregate_values

Правило **allowed_aggregate_values** гарантирует, что любое значение свойства **property_value** совместимо с типом данных этого свойства даже в том случае, если этот тип данных является агрегированным.

Спецификация EXPRESS:

```
    *)
    RULE allowed_aggregate_values FOR (property_value,entity_instance_type_for_aggregate);
    WHERE
    WR1: QUERY (prop <* property_value | NOT (compatible_complete_types_and_value (
        prop.prop_def, prop.its_value))) = [];
    END_RULE; - - allowed_aggregate_values
    (*
```

Формальные положения:

WR1: все значения, ассоциируемые с **property_value** посредством атрибута **its_value**, совместимы с типом данных, который определен атрибутом **prop_def**, ассоциируемым со свойством базовой семантической единицы **property_BSU**, описываемой **property_value**.

7.4 ISO13584_aggregate_value_schema: определение функций

7.4.1 Функция compatible_complete_types_and_value

Функция **compatible_complete_types_and_value** дополняет определенную в ИСО 13584-24 функцию **compatible_type_and_value** проверкой всех типов данных, определенных в данном расширении. Данная функция обеспечивает проверку соответствия агрегированных значений их типу данных.

Если значение, ассоциированное с каким-либо свойством, не является агрегированным, вызывают функцию **compatible_type_and_value**, в противном случае — функцию **compatible_aggregate_domain_and_aggregate_value**.

Спецификация EXPRESS:

```
    *)
    FUNCTION compatible_complete_types_and_value(dom: property_or_data_type_BSU;
        val: primitive_value): LOGICAL;
    IF (data_type_typeof(dom) = [])
    THEN
        RETURN(UNKNOWN);
    END_IF;
    - - checking that values are primitive values but are not aggregate
    - - values.
```

```
IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.PRIMITIVE_VALUE' IN TYPEOF(val))
        AND (NOT('ISO13584_AGGREGATE_VALUE_SCHEMA.' +
        'AGGREGATE_ENTITY_INSTANCE_VALUE' IN TYPEOF(val)))
THEN
        RETURN (compatible_type_and_value(dom, val));
END_IF;
IF 'ISO13584_AGGREGATE_VALUE_SCHEMA.' +
        'AGGREGATE_ENTITY_INSTANCE_VALUE' IN TYPEOF(val)
THEN
        RETURN(compatible_aggregate_domain_and_aggregate_value(dom, val));
END_IF;
- - neither primitive value nor aggregate value
RETURN(UNKNOWN);
END_FUNCTION; - - compatible_complete_types_and_value
(*
```

7.4.2 Функция compatible_aggregate_domain_and_aggregate_value

Функция **compatible_aggregate_domain_and_aggregate_value** обеспечивает проверку того, что элемент **dom property_or_data_type_BSU** имеет правильный тип данных для значения **val agregate_entity_instance_value**.

Эта функция возвращает значение UNKNOWN (неизвестно), если неизвестен конечный тип данных, ассоциируемый с элементом **dom**.

Если в конечном типе данных, ассоциируемом с **dom property_or_data_type_BSU**, нет ссылки на объект **aggregate_type**, функция возвращает значение FALSE (ложно).

Если в конечном типе данных, ассоциируемом с **dom property_or_data_type_BSU**, имеется ссылка на объект **aggregate_type**, то данная функция вызывает функцию **compatible_aggregate_type_and_value**, передавая ей вычисленный тип данных **the_data_type** и значение **val aggregate_entity_instance_value** в качестве параметров.

Спецификация EXPRESS:

```
*)
FUNCTION compatible_aggregate_domain_and_aggregate_value(
        dom: property_or_data_type_BSU;
        val: aggregate_entity_instance_value) : LOGICAL;
LOCAL
        the_data_type: data_type;
END_LOCAL;
- - Check the avalability of the final type of a property or a
- - data type BSU.
IF data_type_typeof(dom) = []
THEN (* the final domain of the type is not available *)
        RETURN(UNKNOWN);
END_IF;
- - Check that the final type of the property or data type BSU
- - is an entity_instance_type whose type_name_attribute
- - references the aggregate_type entity
IF (NOT('ISO13584_IEC61360_DICTIONARY_SCHEMA.ENTITY_INSTANCE_TYPE' IN
        data_type_typeof(dom)))
THEN
        RETURN(FALSE);
END_IF;
IF (NOT('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA' +
'.AGGREGATE_TYPE' IN data_type_type_name(dom)))
THEN
RETURN(FALSE);
END_IF;
- - Compute the final type of the dom property or data_type_BSU
the_data_type := data_type_final_type(dom)[1];
```

```
RETURN(compatible_aggregate_type_and_value(the_data_type, val));
END_FUNCTION; - - compatible_aggregate_domain_and_value
(*
```

7.4.3 Функция data_type_final_type

Функция **data_type_final_type** обеспечивает вычисление типа данных **data_type**, определяющего конечную область значений элементов **property_BSU** или **data_type_BSU**.

Если тип данных **data_type** ассоциирован с именованными типами **named_type**, функция рекурсивно проверяет атрибуты типов **referred_type** до тех пор, пока не обнаружит тип данных, не являющийся простым или сложным (**simple_type** и **complex_type**). После этого функция возвращает набор, содержащий объект **data_type**.

Если определение BSU недоступно, а результат данной функции не может быть сведен к **simple_type** или **complex_type**, функция возвращает пустой набор **data_type**.

Спецификация EXPRESS:

```
*)
FUNCTION data_type_final_type(
        type_spec: property_or_data_type_BSU): SET [0:1] OF data_type;
LOCAL
        res: BOOLEAN := FALSE;
        x: data_type;
END_LOCAL;
IF NOT EXISTS(type_spec) THEN
RETURN([]); - - type_spec is indeterminate
END_IF;
IF ('ISO13584_IEC61360_DICTIONARY_SCHEMA.PROPERTY_BSU' IN
        TYPEOF(type_spec))
THEN
        IF NOT(SIZEOF(type_spec.definition) = 0)
        THEN
            x := type_spec.definition[1]\property_DET.domain;
            res := TRUE;
        END_IF;
ELSE
        IF NOT(SIZEOF(type_spec.definition) = 0)
        THEN
            x := type_spec.definition[1]\data_type_element.
                type_definition;
            res := TRUE;
        END_IF;
END_IF;
IF NOT(res) THEN
        RETURN([]);
END_IF;
IF ('ISO13584_IEC61360_DICTIONARY_SCHEMA.NAMED_TYPE' IN TYPEOF(x)) THEN
        IF NOT(SIZEOF(x\named_type.referred_type.definition) = 0)
    THEN
          RETURN(data_type_final_type(x\named_type.referred_type));
        ELSE
            RETURN([]);
        END_IF;
ELSE
        RETURN([x]);
END_IF;
END_FUNCTION; - - data_type_final_type
(*
```

7.4.4 Функция compatible_aggregate_type_and_value

Функция **compatible_aggregate_type_and_value** обеспечивает проверку совместимости типов **entity_instance_type_for_aggregate** и **aggregate_entity_instance_value**.

Сначала данная функция вычисляет тип элементов, содержащихся в агрегированном значении **val**. Если конечный тип данных агрегированного значения недоступен, функция возвращает значение UNKNOWN (неизвестно). Если типом агрегированного значения (агрегата) является **named_type**, для определения конечного типа элементов вызывают функцию **data_type_final_type**.

При вычислении окончательного типа проверяют каждый из возможных агрегированных типов (список, множество, мультимножество, массив и набор подмножеств).

Для каждого агрегата функция выполняет проверку того, что нижняя и верхняя границы значений соответствуют верхней и нижней границам значений при объявлении типа.

Обработку агрегированного типа данных массива проводят следующим образом.

Для типа данных массива и его значений функция выполняет проверку того, соответствует ли размер списка сохраняемых значений **the_value** размеру, указанному при объявлении типа.

Для списка или массива значений, в которых запрещены повторяющиеся элементы, функция проводит проверку того, чтобы ни одно из значений не повторялось в списке **the_value** дважды. Для массива, в элементах которого не допускается использование значений по умолчанию, функция проводит проверку отсутствия в списке **the_value** неопределенных значений **null_value**.

После этого функция проводит проверку совместимости всех значений из списка **the_value** с соответствующими типами данных элементов **type_of_element**. Для проверки совместимости типов и значений данная функция вызывает функцию **compatible_final_type_and_value** путем рекурсивного неявного вызова.

Спецификация EXPRESS:

```
*)
FUNCTION compatible_aggregate_type_and_value(
        the_data_type: entity_instance_type_for_aggregate;
    val: aggregate_entity_instance_value): LOGICAL;
LOCAL
        elements: LIST OF null_or_primitive_value;
    type_of_elements: data_type;
  result: LOGICAL;
        tmp: LIST OF primitive_value := [];
END_LOCAL;
elements := val.the_value.values;
- - вычисление типа элементов, входящих в агрегат
IF ('ISO13584_IEC61630_DICTIONARY_SCHEMA.NAMED_TYPE' IN
      TYPEOF(the_data_type.type_structure.value_type))
THEN
    IF (data_type_typeof(the_data_type.type_structure.
  value_type\named_type.referred_type) = [])
  THEN
    (* конечная область значений типа неизвестна *)
            RETURN(UNKNOWN);
    END_IF;
  type_of_elements := data_type_final_type(the_data_type.
  type_structure.value_type\named_type.referred_type)[1];
ELSE
  type_of_elements := the_data_type.type_structure.value_type;
END_IF;
- - проверка того, что значение агрегата совместимо с его типом, указанным в объявлении
агрегата
IF ('ISO13584_AGGREGATE_VALUE_SCHEMA.LIST_VALUE'
  IN TYPEOF (val.the_value))
  AND NOT
  ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA'+
  '.LIST_TYPE' IN TYPEOF(the_data_type.type_structure))
```

```
        THEN
          RETURN(FALSE);
        END_IF;
        IF ('ISO13584_AGGREGATE_VALUE_SCHEMA.BAG_VALUE' IN TYPEOF(
          val.the_value)) AND NOT
          ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA.'+
          'BAG_TYPE' IN TYPEOF(the_data_type.type_structure))
        THEN
          RETURN(FALSE);
        END_IF;
        IF ('ISO13584_AGGREGATE_VALUE_SCHEMA.SET_VALUE' IN TYPEOF(
          val.the_value)) AND NOT
          ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA.'+
          'SET_TYPE' IN TYPEOF(the_data_type.type_structure))
        THEN
          RETURN(FALSE);
        END_IF;
        IF ('ISO13584_AGGREGATE_VALUE_SCHEMA.ARRAY_VALUE' IN TYPEOF(
          val.the_value)) AND NOT
          ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA.'+
          'ARRAY_TYPE' IN TYPEOF(the_data_type.type_structure))
        THEN
          RETURN(FALSE);
        END_IF;
        IF
    ('ISO13584_AGGREGATE_VALUE_SCHEMA.SET_WITH_SUBSET_CONSTRAINT_VALUE'
                  IN TYPEOF(val.the_value)) AND NOT
                  ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA.'+
                        'SET_WITH_SUBSET_CONSTRAINT_TYPE'
                      IN TYPEOF(the_data_type.type_structure))
        THEN
          RETURN(FALSE);
        END_IF;
        - - проверка того, что верхняя и нижняя границы значений совместимы с объявлением типа
        IF  (('ISO13584_AGGREGATE_VALUE_SCHEMA.LIST_VALUE'
                    IN TYPEOF (val.the_value))
              OR ('ISO13584_AGGREGATE_VALUE_SCHEMA.BAG_VALUE'
                    IN TYPEOF (val.the_value))
              OR ('ISO13584_AGGREGATE_VALUE_SCHEMA.SET_VALUE'
                    IN TYPEOF (val.the_value))
              OR
        ('ISO13584_AGGREGATE_VALUE_SCHEMA.SET_WITH_SUBSET_CONSTRAINT_VALUE'
                    IN TYPEOF (val.the_value)))
              AND (
          (EXISTS(the_data_type.type_structure.bound_1))
            AND NOT
                (SIZEOF(elements) >= the_data_type.type_structure.bound_1))
        THEN
              RETURN(FALSE);
        END_IF;
        IF (('ISO13584_AGGREGATE_VALUE_SCHEMA.LIST_VALUE' IN TYPEOF(
                    val.the_value))
              OR ('ISO13584_AGGREGATE_VALUE_SCHEMA.BAG_VALUE' IN TYPEOF(
                    val.the_value))
```

```
        OR ('ISO13584_AGGREGATE_VALUE_SCHEMA.SET_VALUE' IN TYPEOF(
            val.the_value))
        OR
('ISO13584_AGGREGATE_VALUE_SCHEMA.SET_WITH_SUBSET_CONSTRAINT_VALUE'
            IN TYPEOF (val.the_value)))
        AND (
      (EXISTS(the_data_type.type_structure.bound_2))
    AND NOT
        (SIZEOF(elements) <= the_data_type.type_structure.bound_2))
THEN
        RETURN(FALSE);
END_IF;
- - для типа данных массива и его значений проверка того, что размер сохраняемого списка
соответствует объявлению типа
IF ('ISO13584_AGGREGATE_VALUE_SCHEMA.ARRAY_VALUE' IN TYPEOF(
        val.the_value))
THEN
        IF (NOT(val.the_value\array_value.bound_1 =
        the_data_type.type_structure.bound_1) OR
            NOT(val.the_value\array_value.bound_2 = the_data_type.type_structure.bound_2))
        THEN
            RETURN(FALSE);
        END_IF;
END_IF;
- - для типа данных set_with_subset_constraint_type
- - проверка того, что ограничения на размер подмножества, определенные
- - на уровне значений, согласованы с соответствующими ограничениями,
- - определенными на уровне типа
IF
('ISO13584_AGGREGATE_VALUE_SCHEMA.SET_WITH_SUBSET_CONSTRAINT_VALUE'
            IN TYPEOF (val.the_value))
    AND NOT
   ((the_data_type.type_structure.cardinal_min
     <= val.the_value\set_with_subset_constraint_value.min)
   AND
     (val.the_value\set_with_subset_constraint_value.min
   <= val.the_value\set_with_subset_constraint_value.max)
     AND
   (val.the_value\set_with_subset_constraint_value.max
     <= the_data_type.type_structure.cardinal_max))
THEN
   RETURN(FALSE);
END_IF;
- - для списка или массива значений с запретом дублирования элементов
- - проверка отсутствия повторяющихся значений
IF ('ISO13584_AGGREGATE_VALUE_SCHEMA.LIST_VALUE' IN TYPEOF(
        val.the_value)) OR
      ('ISO13584_AGGREGATE_VALUE_SCHEMA.ARRAY_VALUE'
   IN TYPEOF (val.the_value))
THEN
        IF (the_data_type.type_structure.uniqueness)
      THEN
          REPEAT i := 1 TO SIZEOF(val.the_value.values);
                IF NOT('ISO13584_INSTANCE_RESOURCE_SCHEMA.NULL_VALUE'
                IN TYPEOF(val.the_value.values[i]))
```

```
                    THEN
                    tmp := tmp + val.the_value.values[i];
                    END_IF;
                END_REPEAT;
            IF NOT(VALUE_UNIQUE(tmp))
            THEN
                RETURN(FALSE);
            END_IF;
        END_IF;
END_IF;
- - для массива с запретом значений по умолчанию
- - проверка отсутствия неопределенных значений (null_value)
IF ('ISO13584_AGGREGATE_VALUE_SCHEMA.ARRAY_VALUE'
        IN TYPEOF(val.the_value))
THEN
        IF NOT(the_data_type.type_structure.are_optional)
    THEN
        REPEAT i := 1 TO SIZEOF(val.the_value.values);
                IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.NULL_VALUE'
                    IN TYPEOF(val.the_value.values[i]))
                THEN
                    RETURN(FALSE);
                END_IF;
            END_REPEAT;
        END_IF;
END_IF;
- - проверка совместимости пар значение/тип для всех
- - элементов, входящих в агрегат
result := TRUE;
REPEAT i := 1 TO SIZEOF(elements);
IF NOT('ISO13584_AGGREGATE_VALUE_SCHEMA.NULL_VALUE'
        IN TYPEOF(elements[i]))
THEN
  IF('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA.'
          + 'ENTITY_INSTANCE_TYPE_FOR_AGGREGATE'
          IN TYPEOF(type_of_elements))
        THEN
            IF ('ISO13584_AGGREGATE_VALUE_SCHEMA.'+
            'AGGREGATE_ENTITY_INSTANCE_VALUE' IN
            TYPEOF(elements[i]))
          THEN
                    result := result AND
                compatible_aggregate_type_and_value(
                type_of_elements, elements[i]);
                ELSE
                    RETURN(FALSE);
                END_IF;
        ELSE
        result := result AND compatible_final_type_and_value(
        type_of_elements, elements[i]);
    END_IF;
  IF NOT(result)
  THEN
            RETURN(FALSE);
      END_IF;
END_IF;
```

```
    END_REPEAT;
    RETURN(result);
    END_FUNCTION; — compatible_aggregate_type_and_value
    (*
```

7.4.5 Функция compatible_final_type_and_value

Функция **compatible_final_type_and_value** обеспечивает выполнение проверки совместимости типов **data_type** и **primitive_value**.

Данная функция проверяет соответствующий тип данных, который ассоциирован с типом **typ**.

Для **integer_value**, **real_value**, **number_value**, **boolean_value**, **translatable_string_value** и **entity_instance_value** управление не передается за пределы данной функции.

Для экземпляров словарных классов (**dic_class_instance**) и значений спецификации уровня (**level_spec_value**) вызываются соответственно функции **compatible_class_and_class** и **compatible_level_type_and_instance**.

Данная функция возвращает значение FALSE (ложно), если ни один из этих типов не является возможным типом параметра **val**.

Спецификация EXPRESS:

```
*)
FUNCTION compatible_final_type_and_value(
        typ: data_type; val: primitive_value): LOGICAL;
LOCAL
        set_string: SET OF STRING := [];
        set_integer: SET OF INTEGER := [];
        code_type: non_quantitative_code_type;
        int_type: non_quantitative_int_type;
END_LOCAL;
IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.INTEGER_VALUE' IN TYPEOF(val))
THEN
        IF (('ISO13584_IEC61360_DICTIONARY_SCHEMA.' +
                'NON_QUANTITATIVE_INT_TYPE' IN TYPEOF(typ)))
THEN
        set_integer := [];
                code_type := typ;
                REPEAT j := 1 TO SIZEOF(code_type.domain.its_values);
                        set_integer := set_integer +
                                code_type.domain.its_values[j].value_code;
                END_REPEAT;
                RETURN(val IN set_integer);
        ELSE
                RETURN(('ISO13584_IEC61360_DICTIONARY_SCHEMA.INT_TYPE'
                        IN TYPEOF(typ)) OR
                        (('ISO13584_IEC61360_DICTIONARY_SCHEMA.NUMBER_TYPE'
                        IN TYPEOF(typ))
                        AND NOT('ISO13584_IEC61360_DICTIONARY_SCHEMA.REAL_TYPE'
                        IN TYPEOF(typ))));
        END_IF;
END_IF;
IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.REAL_VALUE' IN TYPEOF(val))
THEN
        RETURN(('ISO13584_IEC61360_DICTIONARY_SCHEMA.REAL_TYPE'
                IN TYPEOF(typ)) OR
                (('ISO13584_IEC61360_DICTIONARY_SCHEMA.NUMBER_TYPE'
                IN TYPEOF(typ))
                AND NOT('ISO13584_IEC61360_DICTIONARY_SCHEMA.INT_TYPE'
                IN TYPEOF(typ))));
END_IF;
```

```
    IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.NUMBER_VALUE' IN TYPEOF(val))
    THEN
        RETURN('ISO13584_IEC61360_DICTIONARY_SCHEMA.NUMBER_TYPE'
            IN TYPEOF(typ));
    END_IF;
    IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.BOOLEAN_VALUE' IN TYPEOF(val))
    THEN
        RETURN('ISO13584_IEC61360_DICTIONARY_SCHEMA.BOOLEAN_TYPE'
            IN TYPEOF(typ));
    END_IF;
    IF ('ISO13584_INSTANCE_RESOURCE_SCHEMA.TRANSLATABLE_STRING_VALUE' IN
    TYPEOF(val))
    THEN
        IF (('ISO13584_IEC61360_DICTIONARY_SCHEMA' +
            '.NON_QUANTITATIVE_CODE_TYPE') IN TYPEOF(typ))
        THEN
                set_string := [];
                code_type := typ;
                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(('ISO13584_INSTANCE_RESOURCE_SCHEMA.STRING_VALUE'
                    IN TYPEOF(val)) AND (val IN set_string));
        ELSE
            RETURN('ISO13584_IEC61360_DICTIONARY_SCHEMA.STRING_TYPE'
                IN TYPEOF(typ));
        END_IF;
    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(typ)) AND
            (typ\entity_instance_type.type_name <= TYPEOF(val))
        THEN
            RETURN(TRUE);
        ELSE
            RETURN(FALSE);
        END_IF;
    END_IF;
    IF 'ISO13584_INSTANCE_RESOURCE_SCHEMA.DIC_CLASS_INSTANCE'
        IN TYPEOF(val)
    THEN
        IF ('ISO13584_IEC61360_DICTIONARY_SCHEMA.CLASS_INSTANCE_TYPE'
            IN TYPEOF(typ))
        THEN
            RETURN(compatible_class_and_class(typ\class_instance_type.
                domain,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(typ))
        THEN
                RETURN(compatible_level_type_and_instance(
                        typ\level_type.levels,
                        TYPEOF(typ\level_type.value_type), val));
        ELSE
                RETURN(FALSE);
        END_IF;
END_IF;
RETURN(FALSE);
END_FUNCTION; - - compatible_final_type_and_value
(*
*)
END_SCHEMA; - - ISO13584_aggregate_value_schema;
(*
```

8 Интегрированная информационная модель библиотеки LIIM 25

Интегрированная информационная модель библиотеки LIIM 25 соответствует требованиям настоящего стандарта в том случае, если выполняются информационные требования **ISO13584_25_IEC61360_5_liim_schema**, приведенные в 8.1, требования поддержки стандартных данных, установленные в **IEC13584_25_IEC61360_5_conformance_schema**, приведенные в приложении D, требования к поддерживаемым методам реализации, а также применимые требования, указанные в нормативных ссылках.

В любом случае реализация должна поддерживать метод, установленный в ИСО 10303-21. Требования к методам реализации приведены в приложении Е настоящего стандарта.

ISO13584_25_IEC61360_5_liim_schema предусматривает наличие ряда опций, которые поддерживаются реализацией. Эти опции сгруппированы в девять классов соответствия. В каждом классе определены опции, которые могут быть выбраны для реализации. Для соответствия конкретному классу реализации необходима поддержка всех сущностей, типов и связанных ограничений, которые определены для данного класса в **ISO13584_25_IEC61360_5_liim_schema**, а также поддержка стандартных данных, используемых в этом классе.

Примечание 1 — Поддержка стандартных данных, ассоциируемых с классом, обеспечивается путем выполнения общего правила, определенного в **IEC13584_25_IEC61360_5_conformance_schema**.

Предусмотрены следующие классы соответствия.

- класс 1: минимальный набор элементов **dictionary_element** из общей схемы словаря ИСО/МЭК с дополнительными агрегированными типами.

Примечание 2 — Общая схема словаря ИСО/МЭК определяется **ISO13584_IEC61360_dictionary_schema**, соответствующей ИСО 13584-42;

- класс 2: набор элементов **dictionary_element** из расширенной схемы словаря, не содержащий функциональных моделей, функциональных видов и агрегированных типов.

Примечание 3 — Расширенная схема словаря определяется **ISO13584_extended_dictionary_schema**, соответствующей ИСО 13584-24;

- класс 3: набор элементов **dictionary_element** из расширенной схемы словаря, включающий в себя функциональные модели, функциональные виды и агрегированные типы;
- класс 4: класс, идентичный классу 3, но содержащий агрегированные типы, уровень вложения которых ограничен;
- класс 5: набор элементов **dictionary_element** из расширенной схемы словаря, не содержащий функциональных моделей, классов функциональных видов, а также агрегированных типов и значений, но включающий в себя подробное описание расширений **class_extension** для классов библиотеки,

- класс 6: набор элементов **dictionary_element** из расширенной схемы словаря, включающий в себя функциональные модели, функциональные виды, агрегированные типы и значения, а также подробное описание расширений **class_extension** для классов библиотеки;
- класс 7: класс, идентичный классу 3, но содержащий агрегированные типы и значения, уровень вложения которых ограничен;
- класс 10: экземпляры элементов и представлений элементов не имеющие определений и библиотечной структуры;
- класс 11: экземпляры элементов и представлений элементов, дополненные определениями, но не имеющие библиотечной структуры.

Примечание 4 — Для объектов **external_file_protocol**, не принадлежащих набору стандартных данных, определенному в приложении D настоящего стандарта или в другом стандарте комплекса 13584 по протоколу обмена видами, значения атрибутов должны быть предварительно согласованы между отправителем и получателем. Настоящий стандарт не устанавливает требований к определению значений этих атрибутов.

Примечание 5 — В классах соответствия 2—7 и 10—11 интегрированной информационной модели библиотеки LIIM 25 ссылка на файл соответствует ссылке на файл протокола HTTP (**http_file**) только в том случае, если его тип и подтип MIME:
- соответствуют общедоступным спецификациям либо
- ассоциированы с программами для чтения данных файлов, общедоступными в сети Интернет.

Перечень поддерживаемых возможностей различных классов соответствия, входящих в интегрированную информационную модель библиотеки LIIM 25, приведен в таблице 1.

Таблица 1 — Опции соответствия интегрированной информационной модели библиотеки LIIM 25

Возможности	Элементы словаря			Библиотечные спецификации (расширение класса)	Представление экземпляров
Класс соответствия	Словарные определения классов элементов	Словарные определения представлений классов элементов и категорий представлений	Свойства агрегированной структуры		
1	x		x		
2	x				
3	x	x	x		
4	x	x	x		
5	x			x	x
6	x	x	x	x	x
7	x	x	x	x	x
10			x		x
11	x	x	x		x

8.1 ISO13584_25_IEC61360_5_liim_schema: краткий листинг

В настоящем подразделе приведена схема на языке EXPRESS, использующая элементы комплекса стандартов ИСО 10303 по интегрированным ресурсам и комплекса стандартов ИСО 13584 по логическим ресурсам и описанию методологии для задания требований к интегрированной информационной модели библиотеки LIIM 25, определенной в настоящем стандарте.

Примечание 1 — К серии стандартов ИСО 10303 по интегрированным ресурсам относятся части 10303-4x и 10303-1xx. К серии стандартов ИСО 13584 по логическим ресурсам относятся части ИСО 13584-2x, а к серии стандартов ИСО 13584 по методологии описания — части 13584-4x.

Расширенный листинг схемы **ISO13584_25_IEC61360_5_liim_schema** с дополнительными ограничениями, определенными в схеме **IEC13584_25_IEC61360_5_conformance_schema**, приведен в приложении А. Схема **ISO13584_25_IEC61360_5_library_implicit_schema** представляет собой информационную модель библиотек поставщика, в которых ссылки на интегрированную информационную модель библиотеки LIIM 25, соответствующую настоящему стандарту, присутствуют независимо от того, имеются ли в них также ссылки на какие-либо протоколы обмена видами.

Примечание 2 — Настоящий стандарт не распространяется на информационную модель интегрированных библиотек.

Спецификация EXPRESS:

```
*)
SCHEMA ISO13584_25_IEC61360_5_liim_schema;
USE FROM ISO13584_IEC61360_dictionary_schema
    (axis1_placement_type,
    axis2_placement_2d_type,
    axis2_placement_3d_type,
    boolean_type,
    class_BSU,
    class_instance_type,
    class_value_assignment,
    complex_type,
    component_class,
    condition_DET,
    data_type_BSU,
    data_type_element,
    dates,
    dependent_P_DET,
    dic_unit,
    dic_value,
    entity_instance_type,
    identified_document,
    int_currency_type,
    int_measure_type,
    int_type,
    integer_type,
    item_class,
    item_names,
    label_with_language,
    level_type,
    material_class,
    mathematical_string,
    named_type,
    non_dependent_P_DET,
    non_quantitative_code_type,
    non_quantitative_int_type,
    non_si_unit,
    number_type,
    placement_type,
    property_BSU,
    property_DET,
    real_currency_type,
    real_measure_type,
    real_type,
    string_type,
    supplier_BSU,
    supplier_element,
    value_domain);
USE FROM ISO13584_IEC61360_language_resource_schema
    (global_language_assignment,
    present_translations,
    translated_label,
    translated_text);
USE FROM ISO13584_instance_resource_schema
    (null_value,
    primitive_value,
```

```
        null_or_primitive_value,
        simple_value,
        null_or_simple_value,
        number_value,
        null_or_number_value,
        integer_value,
        null_or_integer_value,
        real_value,
        null_or_real_value,
        boolean_value,
        null_or_boolean_value,
        translatable_string_value,
        translated_string_value,
        string_value,
        null_or_translatable_string_value,
        complex_value,
        null_or_complex_value,
        entity_instance_value,
        null_or_entity_instance_value,
        defined_entity_instance_value,
        controlled_entity_instance_value,
        STEP_entity_instance_value,
        PLIB_entity_instance_value,
        property_or_data_type_BSU,
        level_spec_value,
        null_or_level_spec_value,
        int_level_spec_value,
        null_or_int_level_spec_value,
        real_level_spec_value,
        null_or_real_level_spec_value,
        property_value,
        context_dependent_property_value,
        dic_class_instance,
        null_or_dic_class_instance,
        dic_component_instance,
        dic_feature_instance,
        dic_material_instance,
        lib_component_instance,
        lib_feature_instance,
        lib_material_instance,
        dic_f_model_instance,
        lib_f_model_instance);
    USE FROM ISO13584_IEC61360_dictionary_aggregate_extension_schema
        (entity_instance_type_for_aggregate,
        list_type,
        set_type,
        bag_type,
        array_type,
        set_with_subset_constraint_type);
    USE FROM ISO13584_extended_dictionary_schema
        (dictionary,
        dictionary_in_standard_format,
        library_iim_identification,
        view_exchange_protocol_identification,
        representation_type,
        geometric_representation_context_type,
```

```
        representation_reference_type,
        program_reference_type,
        program_library_BSU,
        document_BSU,
        supplier_program_library_relationship,
        class_document_relationship,
        representation_P_DET,
        class_related_dictionary_element,
        program_library_element,
        document_element,
        document_element_with_http_access,
        document_element_with_translated_http_access,
        referenced_document,
        referenced_graphics,
        feature_class,
        functional_model_class,
        fm_class_view_of,
        functional_view_class,
        non_instantiable_functional_view_class,
        view_control_variable_range,
        item_class_case_of,
        component_class_case_of,
        material_class_case_of,
        feature_class_case_of,
        a_posteriori_case_of,
        a_posteriori_view_of);
    USE FROM ISO13584_external_file_schema
        (standard_simple_program_protocol,
        non_standard_simple_program_protocol,
        linked_interface_program_protocol,
        standard_data_protocol,
        non_standard_data_protocol,
        http_protocol,
        program_library_content,
        document_content,
        representation_reference,
        program_reference,
        property_value_external_item,
        message,
        illustration,
        A6_illustration,
        A9_illustration,
        translated_external_content,
        not_translated_external_content,
        not_translatable_external_content,
        language_specific_content,
        external_file_unit,
        http_file,
        http_class_directory,
        simple_program_protocol);
    USE FROM ISO13584_aggregate_value_schema
        (aggregate_entity_instance_value,
        list_value,
        set_value,
        bag_value,
```

```
        array_value,
        set_with_subset_constraint_value);
USE FROM ISO13584_library_content_schema
        (library,
        library_in_standard_format,
        explicit_item_class_extension,
        explicit_functional_model_class_extension,
        property_classification,
        property_value_recommended_presentation);
USE FROM measure_schema
        (amount_of_substance_measure,
        area_measure,
        context_dependent_measure,
        context_dependent_unit,
        conversion_based_unit,
        count_measure,
        derived_unit,
        derived_unit_element,
        dimensional_exponents,
        electric_current_measure,
        global_unit_assigned_context,
        length_measure,
        length_measure_with_unit,
        length_unit,
        luminous_intensity_measure,
        mass_measure,
        measure_value,
        measure_with_unit,
        named_unit,
        numeric_measure,
        parameter_value,
        plane_angle_measure,
        positive_length_measure,
        positive_plane_angle_measure,
        ratio_measure,
        si_unit,
        solid_angle_measure,
        thermodynamic_temperature_measure,
        time_measure,
        volume_measure);
USE FROM person_organization_schema
        (address,
        organization,
        person);
USE FROM date_time_schema
        (date,
        date_and_time,
        local_time,
        calendar_date,
        ordinal_date,
        week_of_year_and_day_date);
USE FROM geometry_schema
        (axis1_placement,
        axis2_placement_2D,
        axis2_placement_3D,
        geometric_representation_context,
        placement);
```

```
USE FROM representation_schema
    (representation,
    representation_context,
    representation_item);
USE FROM application_context_schema
    (application_context,
    application_context_element,
    application_protocol_definition);
END_SCHEMA; - - ISO13584_25_IEC61360_5_liim_schema
(*
```

Примечание — Требования к схемам приведены в следующих документах:

- **ISO13584_IEC61360_dictionary_schema** — МЭК 61360-2 (приложение D ИСО 13584-42);
- **ISO13584_IEC61360_language_resource_schema** — МЭК 61360-2 (приложение D ИСО 13584-42);
- **ISO13584_instance_resource_schema** — ИСО 13584-24;
- **ISO13584_IEC61360_dictionary_aggregate_extension_schema** — настоящий стандарт;
- **ISO13584_extended_dictionary_schema** — ИСО 13584-24;
- **ISO13584_external_file_schema** — ИСО 13584-24;
- **ISO13584_aggregate_value_schema** — настоящий стандарт;
- **ISO13584_library_content_schema** — ИСО 13584-24;
- **measure_schema** — ИСО 10303-41;
- **person_organization_schema** — ИСО 10303-41;
- **date_time_schema** — ИСО 10303-41;
- **geometry_schema** — ИСО 10303-42;
- **representation_schema** — ИСО 10303-43;
- **application_context_schema** — ИСО 10303-41.

8.2 Требования к классам соответствия

8.2.1 Класс соответствия 1: минимальные словари

Класс соответствия 1 распространяется на реализации, предназначенные для поддерживания общих требований, установленных в схеме словаря ИСО/МЭК, а также в ее расширениях, в которых установлены значения и требования к агрегированным типам данных. Реализация класса соответствия 1 интегрированной информационной модели библиотеки LIIM 25 должна поддерживать следующие сущности и связанные с ними конструкции:

```
FROM ISO13584_IEC61360_dictionary_schema
    supplier_BSU
    supplier_element
    class_BSU
    item_class
    component_class
    material_class
    property_BSU
    property_DET
    condition_DET
    dependent_P_DET
    non_dependent_P_DET
    class_value_assignment
    data_type_BSU
    data_type_element
    number_type
    int_type
    int_measure_type
    int_currency_type
    integer_type
    non_quantitative_int_type
    real_type
    real_measure_type
    real_currency_type
```

boolean_type
string_type
non_quantitative_code_type
complex_type
level_type
class_instance_type
entity_instance_type
placement_type
axis1_placement_type
axis2_placement_2d_type
axis2_placement_3d_type
named_type
value_domain
dic_value
non_si_unit
dic_unit
dates
identified_document
item_names
label_with_language
mathematical_string
FROM ISO13584_IEC61360_language_resource_schema
global_language_assignment
present_translations
translated_label
translated_text
FROM ISO13584_IEC61360_dictionary_aggregate_extension_schema
aggregate_entity_instance_type
list_type
set_type
bag_type
array_type
set_with_subset_constraint_type
FROM measure_schema
amount_of_substance_measure
area_measure
context_dependent_measure
context_dependent_unit
conversion_based_unit
count_measure
derived_unit
derived_unit_element
dimensional_exponents
electric_current_measure
global_unit_assigned_context
length_measure
length_measure_with_unit
length_unit
luminous_intensity_measure
mass_measure
measure_value
measure_with_unit
named_unit
numeric_measure
parameter_value

plane_angle_measure
positive_length_measure
positive_plane_angle_measure
ratio_measure
si_unit
solid_angle_measure
thermodynamic_temperature_measure
time_measure
volume_measure

FROM person_organization_schema
address
organization

8.2.2 Класс соответствия 2: словари классов элементов

Класс соответствия 2 распространяется на реализации, поддерживающие элементы словаря **dictionary_element**, расширенной схемы словаря, не содержащей классов функциональных моделей и функциональных видов, а также агрегированных типов данных. Реализация класса соответствия 2 интегрированной информационной модели библиотеки LIIM 25 должна поддерживать следующие сущности и связанные с ними конструкции:

FROM ISO13584_IEC61360_dictionary_schema
supplier_BSU
supplier_element
class_BSU
item_class
component_class
material_class
property_BSU
property_DET
condition_DET
dependent_P_DET
non_dependent_P_DET
class_value_assignment
data_type_BSU
data_type_element
number_type
int_type
int_measure_type
int_currency_type
integer_type
non_quantitative_int_type
real_type
real_measure_type
real_currency_type
boolean_type
string_type
non_quantitative_code_type
complex_type
level_type
class_instance_type
entity_instance_type
placement_type
axis1_placement_type
axis2_placement_2d_type
axis2_placement_3d_type
named_type
value_domain
dic_value

non_si_unit
dic_unit
dates
identified_document
item_names
label_with_language
mathematical_string
FROM ISO13584_IEC61360_language_resource_schema
global_language_assignment
present_translations
translated_label
translated_text
FROM ISO13584_extended_dictionary_schema
dictionary
dictionary_in_standard_format
library_iim_identification
view_exchange_protocol_identification
document_BSU
class_document_relationship
representation_P_DET
class_related_dictionary_element
document_element
document_element_with_http_access
documented_element_with_translated_http_access
referenced_document
referenced_graphics
feature_class
item_class_case_of
component_class_case_of
material_class_case_of
feature_class_case_of
a_posteriori_case_of
a_posteriori_view_of
FROM ISO13584_external_file_schema
standard_data_protocol
non_standard_data_protocol
http_protocol
document_content
translated_external_content
not_translated_external_content
not_translatable_external_content
language_specific_content
external_file_unit
http_file
http_class_directory
simple_program_protocol
FROM measure_schema
amount_of_substance_measure
area_measure
context_dependent_measure
context_dependent_unit
conversion_based_unit
count_measure
derived_unit

derived_unit_element
dimensional_exponents
electric_current_measure
global_unit_assigned_context
length_measure
length_measure_with_unit
length_unit
luminous_intensity_measure
mass_measure
measure_value
measure_with_unit
named_unit
numeric_measure
parameter_value
plane_angle_measure
positive_length_measure
ratio_measure
si_unit
solid_angle_measure
thermodynamic_temperature_measure
time_measure
volume_measure

FROM person_organization_schema

address
organization

8.2.3 Класс соответствия 3: полные словари

Класс соответствия 3 распространяется на реализации, поддерживающие как класс соответствия 2, так и классы функциональных моделей, классы функциональных видов, а также агрегированные типы данных. Реализация класса соответствия 3 интегрированной информационной модели библиотеки LIIM 25 должна поддерживать все сущности, поддерживаемые классом соответствия 2, а также следующие сущности и связанные с ними конструкции:

FROM ISO13584_extended_dictionary_schema

representation_type
geometric_representation_context_type
representation_reference_type
supplier_program_library_relationship
functional_model_class
fm_class_view_of
functional_view_class
non_instantiable_functional_view_class
view_control_variable_range

FROM ISO13584_external_file_schema

standard_simple_program_protocol
non_standard_simple_program_protocol
linked_interface_program_protocol
program_library_content
representation_reference
program_reference

FROM ISO13584_IEC61360_dictionary_aggregate_extension_schema

aggregate_entity_instance_type
list_type
set_type
bag_type
array_type
set_with_subset_constraint_type

8.2.4 Класс соответствия 4: полные словари с ограниченным уровнем вложения агрегированных значений

Класс соответствия 4 распространяется на реализации, поддерживающие все сущности и связанные с ними конструкции, которые определены для класса соответствия 3 с учетом того, что уровень вложения агрегатов в этих реализациях ограничен значением 2 в соответствии с правилом **nesting_level_aggregate_limit_rule**, определенным в приложении D.

8.2.5 Класс соответствия 5: библиотеки классов элементов

Класс соответствия 5 распространяется на реализации, поддерживающие класс соответствия 2, а также описание расширений элементов классов путем определения набора их экземпляров. Класс соответствия 5 не поддерживает экземпляры функциональных моделей и значения агрегированной структуры. Реализация класса соответствия 5 интегрированной информационной модели библиотеки LIIM 25 должна поддерживать объекты класса соответствия 2, а также следующие сущности и связанные с ними конструкции:

FROM ISO13584_extended_dictionary_schema
FROM ISO13584_external_file_schema
property_value_external_item,
message,
illustration,
A6_illustration,
A9_illustration,
FROM ISO13584_instance_resource_schema
null_value
primitive_value
null_or_primitive_value
simple_value
null_or_simple_value
number_value
null_or_number_value
integer_value
null_or_integer_value
real_value
null_or_real_value
boolean_value
null_or_boolean_value
translatable_string_value
translated_string_value
string_value
null_or_translatable_string_value
complex_value
null_or_complex_value
entity_instance_value
null_or_entity_instance_value
defined_entity_instance_value
controlled_entity_instance_value
STEP_entity_instance_value
PLIB_entity_instance_value
property_or_data_type_BSU
level_spec_value
null_or_level_spec_value
Int_level_spec_value
null_or_int_level_spec_value
real_level_spec_value
null_or_real_level_spec_value
property_value
context_dependent_property_value
dic_class_instance

null_or_dic_class_instance
dic_component_instance
dic_feature_instance
dic_material_instance
lib_component_instance
lib_feature_instance
lib_material_instance
FROM ISO13584_library_content_schema
library
library_in_standard_format
explicit_item_class_extension
property_classification
property_value_recommended_presentation
FROM person_organization_schema
person
FROM date_time_schema
date
date_and_time
local_time
calendar_date
ordinal_date
week_of_year_and_day_date
FROM geometry_schema
axis1_placement
axis2_placement_2D
axis2_placement_3D
geometric_representation_context
placement
FROM representation_schema
representation
representation_context
representation_item
FROM application_context_schema
application_context
application_context_element
application_protocol_definition

8.2.6 Класс соответствия 6: полные библиотеки

Класс соответствия 6 распространяется на реализации, поддерживающие класс соответствия 5, а также описание функциональных моделей классов расширения и использование значений агрегированной структуры. Реализация класса соответствия 6 интегрированной информационной модели библиотеки LIIM 25 должна поддерживать сущности, поддерживаемые классом соответствия 5, а также следующие сущности и связанные с ними конструкции:

FROM ISO13584_extended_dictionary_schema
representation_type
geometric_representation_context_type
representation_reference_type
supplier_program_library_relationship
program_reference_type
program_library_BSU
program_library_element
functional_model_class
fm_class_view_of
functional_view_class
non_instantiable_functional_view_class
view_control_variable_range
a_posteriori_view_of

FROM ISO13584_external_file_schema
standard_simple_program_protocol
non_standard_simple_program_protocol
linked_interface_program_protocol
program_library_content
representation_reference
program_reference
FROM ISO13584_instance_resource_schema
dic_f_model_instance
lib_f_model_instance
FROM ISO13584_library_content_schema
explicit_functional_model_class_extension
FROM ISO13584_IEC61360_dictionary_aggregate_extension_schema
entity_instance_type_for_aggregate
list_type
set_type
bag_type
array_type
set_with_subset_constraint_type
FROM ISO13584_aggregate_value_schema
aggregate_entity_instance_value
list_value
set_value
bag_value
array_value
FROM person_organization_schema
person
FROM date_time_schema
date
date_and_time
local_time
calendar_date
ordinal_date
week_of_year_and_day_date

8.2.7 Класс соответствия 7: полные библиотеки с ограниченным уровнем вложения агрегированных значений

Класс соответствия 7 распространяется на реализации, поддерживающие все сущности и связанные с ними конструкции, которые определены для класса соответствия 6 с учетом того, что уровень вложения агрегатов в этих реализациях ограничен значением 2 в соответствии с правилом **nesting_level_aggregate_limit_rule**, определенным в приложении D.

8.2.8 Класс соответствия 10: экземпляры библиотек

Класс соответствия 10 распространяется на реализации, поддерживающие описания экземпляров элементов классов или экземпляров представлений элементов без словарных определений и библиотечной структуры. Реализация класса соответствия 10 интегрированной информационной модели библиотеки LIIM 25 должна поддерживать все следующие сущности и связанные с ними конструкции.

Примечание — Для класса соответствия 10 использование каких-либо объектов словарей или библиотек (**dictionary** или **library**) не является обязательным, если для представления экземпляров не используются протоколы обмена видами.

FROM ISO13584_IEC61360_dictionary_schema
supplier_BSU
supplier_element
class_BSU
property_BSU
data_type_BSU
dic_value

dates
identified_document
item_names
label_with_language
mathematical_string
FROM ISO13584_IEC61360_language_resource_schema
global_language_assignment
present_translations
translated_label
translated_text
FROM ISO13584_instance_resource_schema
null_value
primitive_value
null_or_primitive_value
simple_value
null_or_simple_value
number_value
null_or_number_value
integer_value
null_or_integer_value
real_value
null_or_real_value
boolean_value
null_or_boolean_value
translatable_string_value
translated_string_value
string_value
null_or_translatable_string_value
complex_value
null_or_complex_value
entity_instance_value
null_or_entity_instance_value
defined_entity_instance_value
controlled_entity_instance_value
STEP_entity_instance_value
PLIB_entity_instance_value
property_or_data_type_BSU
level_spec_value
null_or_level_spec_value
Int_level_spec_value
null_or_int_level_spec_value
real_level_spec_value
null_or_real_level_spec_value
property_value
context_dependent_property_value
dic_class_instance
null_or_dic_class_instance
dic_component_instance
dic_feature_instance
dic_material_instance
lib_component_instance
lib_feature_instance
lib_material_instance
dic_f_model_instance
lib_f_model_instance

FROM ISO13584_extended_dictionary_schema
dictionary
dictionary_in_standard_format
library_iim_identification
view_exchange_protocol_identification
program_library_BSU
document_element
document_element_with_http_access
documented_element_with_translated_http_access
referenced_document
referenced_graphics
document_BSU
class_document_relationship
FROM ISO13584_external_file_schema
http_protocol
document_content
translated_external_content
not_translated_external_content
not_translatable_external_content
language_specific_content
external_file_unit
http_file
property_value_external_item
FROM ISO13584_aggregate_value_schema
aggregate_entity_instance_value
list_value
set_value
bag_value
array_value
set_with_subset_constraint_value
FROM ISO13584_library_content_schema
library
library_in_standard_format
FROM person_organization_schema
address
organization
person
FROM date_time_schema
date
date_and_time
local_time
calendar_date
ordinal_date
week_of_year_and_day_date

8.2.9 Класс соответствия 11: экземпляры библиотек с заданными словарными определениями

Класс соответствия 11 распространяется на реализации, поддерживающие описания экземпляров элементов классов или экземпляров представлений элементов, которые дополнены словарными определениями, но не имеют библиотечной структуры. Реализация класса соответствия 11 интегрированной информационной модели библиотеки LIIM 25 должна поддерживать все сущности, поддерживаемые классом соответствия 10, а также следующие сущности и связанные с ними конструкции.

Примечание — Для класса соответствия 11 использование каких-либо сущностей словарей или библиотек (**dictionary** или **library**) не является обязательным, если для представления экземпляров не используются протоколы обмена видами.

```
FROM ISO13584_IEC61360_dictionary_schema
    item_class
    component_class
    material_class
    property_DET
    condition_DET
    dependent_P_DET
    non_dependent_P_DET
    class_value_assignment
    data_type_element
    number_type
    int_type
    int_measure_type
    int_currency_type
    integer_type
    non_quantitative_int_type
    real_type
    real_measure_type
    real_currency_type
    boolean_type
    string_type
    non_quantitative_code_type
    complex_type
    level_type
    class_instance_type
    entity_instance_type
    placement_type
    axis1_placement_type
    axis2_placement_2d_type
    axis2_placement_3d_type
    named_type
    value_domain
    non_si_unit
    dic_unit
FROM ISO13584_extended_dictionary_schema
    representation_P_DET
    class_related_dictionary_element
    feature_class
    item_class_case_of
    component_class_case_of
    material_class_case_of
    feature_class_case_of
    a_posteriori_case_of
    a_posteriori_view_of
    representation_type
    geometric_representation_context_type
    representation_reference_type
    supplier_program_library_relationship
    functional_model_class
    fm_class_view_of
    functional_view_class
    non_instantiable_functional_view_class
    view_control_variable_range
FROM ISO13584_external_file_schema
    standard_data_protocol
    non_standard_data_protocol
    http_class_directory
```

simple_program_protocol
standard_simple_program_protocol,
non_standard_simple_program_protocol,
linked_interface_program_protocol
representation_reference
program_reference
FROM measure_schema
amount_of_substance_measure
area_measure
context_dependent_measure
context_dependent_unit
conversion_based_unit
count_measure
derived_unit
derived_unit_element
dimensional_exponents
electric_current_measure
length_measure
length_measure_with_unit
length_unit
luminous_intensity_measure
mass_measure
measure_value
measure_with_unit
named_unit
numeric_measure
parameter_value
plane_angle_measure
positive_length_measure
positive_plane_angle_measure
ratio_measure
si_unit
solid_angle_measure
thermodynamic_temperature_measure
time_measure
volume_measure
FROM ISO13584_IEC61360_dictionary_aggregate_extension_schema
aggregate_entity_instance_type
list_type
set_type
bag_type
array_type
set_with_subset_constraint_type
FROM geometry_schema
axis1_placement
axis2_placement_2D
axis2_placement_3D
geometric_representation_context
placement
FROM representation_schema
representation
representation_context
representation_item
FROM application_context_schema
application_context
application_context_element
application_protocol_definition

Приложение А
(обязательное)

Сокращенные наименования сущностей, приведенных в настоящем стандарте

Сокращенные наименования сущностей, приведенных в настоящем стандарте, приведены в таблице А.1. Требования к сокращенным наименованиям установлены в описаниях методов реализации ИСО 10303.

Перечни наименований сущностей языка EXPRESS и соответствующих сокращений, предназначенных для использования в схеме на языке EXPRESS, определенной в настоящем стандарте, приведены в сети Интернет по адресу:

http://www.tc184-sc4.org/Short_Names/

Примечание — Информация, приведенная по вышеуказанному адресу, является справочной.

Таблица А.1 — Сокращенные наименования

Полное наименование	Сокращение
AGGREGATE_ENTITY_INSTANCE_VALUE	AEIV
AGGREGATE_TYPE	AGGTYP
AGGREGATE_VALUE	AGGVL
ARRAY_TYPE	ARRTYP
ARRAY_VALUE	ARRVL
BAG_TYPE	BGTYP
BAG_VALUE	BGVL
ENTITY_INSTANCE_TYPE_FOR_AGGREGATE	EITFA
LIST_TYPE	LSTTYP
LIST_VALUE	LSTVL
SET_TYPE	STTYP
SET_WITH_SUBSET_CONSTRAINT_TYPE	SWSCT
SET_VALUE	STVL
SET_WITH_SUBSET_CONSTRAINT_VALUE	SWSCV

Приложение В
(обязательное)

Регистрация информационных объектов

В.1 Идентификация документа

Для обеспечения точной идентификации информационного объекта в открытой системе настоящему стандарту присвоен следующий идентификатор:

{ ISO standard 13584 part (25) version(1) }

Данное значение определено в ИСО/МЭК 8824-1 и описано в ИСО 13584-1.

В.2 Идентификация схемы

В.2.1 ISO13584_IEC61360_dictionary_aggregate_extension_schema

ISO13584_IEC61360_dictionary_aggregate_extension_schema (см. раздел 6) присвоен следующий идентификатор объекта:

{ISO standard 13584 part (25) version(1) object(1) ISO13584-IEC61360-dictionary-aggregate-extension-schema (1)}

В.2.2 ISO13584_aggregate_value_schema

ISO13584_aggregate_value_schema (см. раздел 7) присвоен следующий идентификатор объекта:

{ISO standard 13584 part (25) version(1) object(1) ISO13584-aggregate-value-schema (2) }

В.2.3 Схема ISO13584_25_IEC61360_5_liim_schema

Схеме **ISO13584_25_IEC61360_5_liim_schema** (см. подраздел 8.1) присвоен следующий идентификатор объекта:

{ISO standard 13584 part (25) version(1) object(1) ISO13584-25-liim-schema (3) }

Приложение С
(справочное)

Расширенный листинг схемы ISO13584_25_IEC61360_5_library_implicit_schema

В настоящем приложении приведены ссылки на полный листинг схем на языке EXPRESS, приведенных в разделе 8 настоящего стандарта без комментариев и каких-либо пояснений, но с указанием дополнительных ограничений, установленных в приложении D для **ISO13584_25_IEC61360_5_conformance_schema**. Данная схема имеет наименование **ISO13584_25_IEC61360_5_library_implicit_schema**. В листинге этой схемы указаны все элементы соответствующей краткой схемы по разделу 8. Эти элементы заимствованы из других схем и объединены в одну схему без каких-либо ссылок на внешние источники.

Данная схема может использоваться:

- для обмена библиотеками, содержащими ссылки на **ISO13584_25_IEC61360_5_liim_schema** и ассоциированную с ней **IEC13584_25_IEC61360_5_conformance_schema**, но не на какой-либо протокол обмена видами;

- для обмена библиотеками, содержащими ссылки на **ISO13584_25_IEC61360_5_liim_schema** и ассоциированную с ней **IEC13584_25_IEC61360_5_conformance_schema**, а также ссылки на некоторые протоколы обмена видами. В этом случае ограничения, установленные в используемых протоколах обмена видами, проверке не подвергают.

Данная схема может быть также дополнена проверкой ограничений, установленных во всех используемых протоколах обмена видами. Для каждого протокола, на который имеется ссылка, эту проверку проводят следующим образом, предполагая, что имеется ссылка на протокол обмена видами V1, и что в этом протоколе установлены две схемы ограничений с именами S1_V1 и S2_V1:

a) проводят проверку того, что сущности, к которым происходит обращение в схемах S1_V1 и S2_V1, уже существуют в **ISO13584_25_IEC61360_5_library_implicit_schema**. В противном случае ссылки на интегрированную информационную модель библиотеки LIIM 25 и на протокол обмена видами V1 из одного файла поставки библиотеки недопустимы.

Примечание 1 — Информационная модель файла поставки библиотеки и сущности, которые могут в него входить, определены в интегрированной информационной модели библиотеки. Протокол обмена видами может добавлять только ограничения;

b) конструируют расширенную форму схемы S1_V1 и результату присваивают наименование "S1_V1";

c) конструируют расширенную форму схемы S2_V1 и результату присваивают наименование "S2_V1";

d) во всех местах расширенной формы схемы S1_V1, в которых имеется строка "S1_V1", эту строку заменяют на **'ISO13584_25_IEC61360_5_library_implicit_schema'** с сохранением контекста;

e) во всех местах расширенной формы схемы S2_V1, в которых имеется строка "S2_V1", эту строку заменяют на **'ISO13584_25_IEC61360_5_library_implicit_schema'** с сохранением контекста;

f) к содержимому **ISO13584_25_IEC61360_5_library_implicit_schema** добавляют содержимое расширенной формы схемы S1_V1 и удаляют повторяющиеся элементы;

g) к содержимому **ISO13584_25_IEC61360_5_library_implicit_schema** добавляют содержимое расширенной формы схемы S2_V1 и удаляют повторяющиеся элементы.

После выполнения данной процедуры для всех протоколов обмена видами V1, V2,..., Vn **ISO13584_25_IEC61360_5_library_implicit_schema** может использоваться для обмена любыми библиотеками, использующими **ISO13584_25_IEC61360_5_liim_schema** и соответствующую ей **IEC13584_25_IEC61360_5_conformance_schema** в качестве интегрированной информационной модели и содержащими ссылки на весь набор протоколов обмена видами V1, V2,..., Vn либо на его часть. В окончательную схему кроме того должны входить ограничения, установленные в протоколах обмена видами, ссылки на которые содержатся в ней.

Листинг **ISO13584_25_IEC61360_5_library_implicit_schema** приведен в сети Интернет по адресу:

gost.e-ecolog.ru /EXPRESS/

В случае затруднений, связанных с доступом на сайт, обращайтесь непосредственно в центральный секретариат ИСО или в секретариат ИСО ТК 184/ПК4 по адресу электронной почты: gost.e-ecolog.ru

Примечание 2 — Информация, приведенная по указанному выше адресу в виде, пригодном для ее интерпретации компьютером, является нормативной.

Примечание 3 — Информация о выявленных ошибках в коде EXPRESS, описание этих ошибок, а также изменениях доступна в сети Интернет по адресу:

gost.e-ecolog.ru

Приложение D
(обязательное)

Требования к стандартным данным интегрированной информационной модели библиотеки LIIM 25

Стандартные данные являются экземплярами сущностей, которые должны распознаваться любой реализацией, соответствующей требованиям комплекса стандартов ИСО 13584 и заявляющей о соответствии определенному классу соответствия конкретной интегрированной информационной модели библиотеки или какому-либо протоколу обмена представлениями ИСО 13584.

Стандартные данные должны определяться каждой интегрированной информационной моделью библиотеки и каждым протоколом обмена видами. Для каждого класса соответствия все элементы из других схем, которые используются в соответствующей краткой форме схемы, приведенной в разделе 8, в каждом расширенном листинге по настоящему приложению объединены в одну схему, не содержащую внешних ссылок ни на одну из этих схем.

Стандартными данными могут быть:

- экземпляры базовых семантических единиц **basic_semantic_unit**, ассоциируемые с соответствующими элементами словаря **dictionary_element** и, возможно, с элементом содержимого **content_item**;
- экземпляры протоколов из внешних файлов **external_file_protocol**, а также
- экземпляры других объектов, необходимых для определения выше указанных экземпляров.

Распознавание полученной базовой семантической единицы **basic_semantic_unit** означает, что соответствующая единица **basic_semantic_unit** уже имеется в библиотеке пользователя вместе с соответствующим элементом **dictionary_element** и, возможно, с элементом **content_item** в виде стандартных данных протокола обмена видами и интегрированной информационной модели библиотеки. Вследствие этого любую ссылку на равнозначное значение **basic_semantic_unit** в библиотеке поставщика интерпретируют в виде ссылки на ранее заданную единицу **basic_semantic_unit**.

Примечание 1 — Примерами базовых семантических единиц, которые могут быть определены как стандартные данные в протоколе обмена видами, являются единицы класса функционального вида **class_BSU**, который может быть указан в протоколе обмена видами, а также единицы **property_BSU**, которые обозначают переменные управления видом данного класса функционального вида.

Распознавание внешнего файла протокола означает, что внешние файлы, ссылающиеся на протокол **external_file_protocol** аналогичного значения, должны обрабатываться реализацией, которая поддерживает данный протокол.

Примечание 2 — Примером протокола внешнего файла, который может быть указан в качестве стандартных данных в протоколе обмена видами или интегрированной информационной модели библиотеки, является ИСО 8859-1, устанавливающий требования к 8-битовому кодированию набора графических символов латинского алфавита № 1.

Стандартные данные устанавливают путем введения набора ограничений, которые должны выполняться любой библиотекой, претендующей на соответствие некоторому классу соответствий LIIM 25. Следующие стандартные данные установлены в интегрированной информационной модели библиотеки LIIM 25.

D.1 Допустимые ссылки из файла поставки библиотеки на интегрированную информационную модель библиотеки LIIM 25

В данном разделе указаны значения экземпляров **library_iim_identification**, которые могут быть использованы в файле поставки библиотеки для ссылок на интегрированную информационную модель библиотеки LIIM 25, соответствующую настоящему стандарту.

Допустимые значения приведены в таблице D.1, в которой для каждого класса соответствия указаны допустимые значения **library_iim_identification.name** и **library_iim_identification.application**, и в схеме на языке EXPRESS, которая устанавливает общие правила. Это правило должно быть выполнено для любого файла поставки библиотеки, в котором имеются ссылки на интегрированную информационную модель библиотеки LIIM 25, определенную в настоящем стандарте для всех классов соответствия. Данное правило устанавливает допустимые значения других атрибутов **library_iim_identification**, которые следует использовать для ссылок на интегрированную информационную модель библиотеки LIIM 25 путем задания отношений с **view_exchange_protocol_identification.name** и **view_exchange_protocol_identification.application**.

Данное правило включено в схему **ISO13584_25_IEC61360_5_library_implicit_schema**, приведенную в разделе D.4.1.

D.2 Таблица спецификаций классов соответствия

В таблице D.1 приведены значения **library_iim_identification.name** и **library_iim_identification.application**, которые могут быть использованы в **library_iim_identification** для ссылок на интегрированную информационную модель библиотеки LIIM 25 в любом из ее классов соответствия.

Таблица D.1 — Спецификация классов соответствия модели LIIM 25 по ИСО 13584

Класс соответствия	Обязательное значение library_iim_identification.name	Обязательное значение library_iim_identification.application
2	'ИСО13584 25 МЭК61360 5'	'2'
3	'ИСО13584 25 МЭК61360 5'	'3'
4	'ИСО13584 25 МЭК61360 5'	'4'
5	'ИСО13584 25 МЭК61360 5'	'5'
6	'ИСО13584 25 МЭК61360 5'	'6'
7	'ИСО13584 25 МЭК61360 5'	'7'
10	'ИСО13584 25 МЭК61360 5'	'10'
11	'ИСО13584 25 МЭК61360 5'	'11'

Примечание 1 — Класс соответствия 1 не указан, так как в нем отсутствуют ссылки на тип данных сущности **library_iim_identification**, поэтому для этого класса не приведены стандартные данные. Классы соответствия 10 и 11 указаны, так как в них допускаются ссылки на тип данных сущности **library_iim_identification**.

Примечание 2 — Правила **allowed_reference_to_LIIM_25_rule**, **allowed_entity_instance_type_in_LIIM_25_rule** и **allowed_language_assignement_rule** применимы для всех элементов, прямо или косвенно используемых в определениях классов соответствий 1—7 и 10—11. Однако правило **allowed_reference_to_LIIM_25_rule** не имеет значения в случае класса соответствия 1 и может не действовать в случае классов 10 и 11, так как в случае двух последних классов **library_iim_identification** может не использоваться.

D.3 Стандартные данные для всех классов соответствия, кроме класса 1

Настоящий раздел устанавливает ограничения для файла поставки библиотеки, соответствующего интегрированной информационной модели библиотеки LIIM 25.

Значения экземпляров **library_iim_identification**, которые могут использоваться в файле поставки библиотеки, соответствующем интегрированной информационной модели библиотеки LIIM 25, соответствующей настоящему стандарту для классов соответствия 2—7 и 10—11, должны учитывать ограничения, установленные в следующей схеме на языке EXPRESS.

Спецификация EXPRESS:

```
*)
SCHEMA ISO13584_25_IEC61360_5_conformance_schema;
USE FROM ISO13584_IEC61360_language_resource_schema(
        translated_label,
        present_translations,
        global_language_assignment);
USE FROM ISO13584_IEC61360_dictionary_aggregate_extension_schema(
        aggregate_type,
        entity_instance_type);
USE FROM ISO13584_extended_dictionary_schema(
        data_exchange_specification_identification,
        library_iim_identification);
USE FROM ISO13584_external_file_schema(
        external_file_protocol);
(*
```

Примечание — Требования к данной схеме установлены в следующих документах:

- **ISO13584_IEC61360_language_resource_schema** — МЭК 61360-2;
- **ISO13584_IEC61360_dictionary_schema** — МЭК 61360-2;
- **ISO13584_extended_dictionary_schema** — ИСО 13584-24;
- **ISO13584_external_file_schema** — ИСО 13584-24.

D.3.1 Правило allowed_reference_to_LIIM_25_rule

Правило **allowed_reference_to_LIIM_25_rule** устанавливает формальное и неформальное ограничения к сущностям **library_iim_identification**, которые могут быть использованы для ссылок на классы соответствия 1—7 и 10—11 интегрированной информационной модели библиотеки LIIM 25, соответствующей настоящему стандарту. **library_iim_identification** допускается использовать для ссылок на классы соответствия 1—7 и 10—11 интегрированной информационной модели библиотеки LIIM 25, если выполняются следующие условия:

- атрибут **name** сущности **library_iim_identification**, который ссылается на интегрированную модель LIIM 25, равен 'ISO13584_25_IEC61360_5';

- атрибут **status** сущности **library_iim_identification** имеет одно из следующих значений: 'WD', 'CD', 'DIS', 'FDIS', 'IS', 'TS', 'PAS' или 'ITA';

- атрибут **application** сущности **library_iim_identification** имеет значение '2', '3', '4', '5', '6', '7', '10' или '11', а значения **external_file_protocol**, на которые есть ссылки в атрибуте **external_file_protocols** сущности **library_iim_identification**, соответствуют ограничениям, определенным функцией **compliant_external_file_protocol_25**.

Кроме того, **library_iim_identification** допустимо использовать для ссылок на классы соответствия 1—7 и 10—11 интегрированной информационной модели библиотеки LIIM 25, если выполняется любое из двух следующих условий, относящихся к файлам **http_file**, на которые могут быть прямые или косвенные ссылки из **library_iim_identification**:

- каждому из файлов **http_file**, на которые имеются ссылки, присвоены атрибуты **mime** и **exchange_format**, соответствующие типу и подтипу MIME и описанные в общедоступной спецификации;

- каждому из файлов **http_file** присвоены атрибуты **mime** и **exchange_format**, соответствующие типу и подтипу MIME, приведенные в спецификации программ для чтения этих файлов, доступной в сети Интернет.

Настоящий стандарт не распространяется на ссылки на файлы **http_file**, соответствующие другим типам и подтипам MIME, которые допускаются в случае заключения отдельного соглашения между отправителем и получателем. Это задокументировано в неформальном положении **IP1** правила **allowed_reference_to_LIIM_25_rule**.

Спецификация EXPRESS:

```
*)
RULE allowed_reference_to_LIIM_25_rule FOR (library_iim_identification);
WHERE
    WR1: QUERY( liim_id <* library_iim_identification |
        ((liim_id\data_exchange_specification_identification.status = 'WD') OR
        (liim_id\data_exchange_specification_identification.status = 'CD') OR
        (liim_id\data_exchange_specification_identification.status = 'DIS') OR
        (liim_id\data_exchange_specification_identification.status = 'FDIS') OR
        (liim_id\data_exchange_specification_identification.status = 'IS') OR
        (liim_id\data_exchange_specification_identification.status = 'TS') OR
        (liim_id\data_exchange_specification_identification.status = 'PAS') OR
        (liim_id\data_exchange_specification_identification.status = 'ITA'))
        AND
        (liim_id\data_exchange_specification_identification.name =
        'ISO13584_25_IEC61360_5')
        AND
        is_correct_liim_25_application_value(liim_id)
        AND
        (QUERY( efp <* liim_id\data_exchange_specification_identification
        .external_file_protocols | NOT(compliant_external_file_protocol_25([efp]))) = []))
        = QUERY( liim_id <* library_iim_identification |
        (liim_id\data_exchange_specification_identification.name =
        'ISO13584_25_IEC61360_5'));
END_RULE; - - allowed_reference_to_LIIM_25_rule
*)
```

Формальное положение:

WR1: при ссылках на интегрированную информационную модель библиотеки LIIM 25, соответствующую настоящему стандарту, в качестве значения **library_iim_identification.name** следует использовать 'ISO13584_25_IEC61360_5', значение **library_iim_identification.status** должно быть равно 'WD', 'CD', 'DIS', 'FDIS', 'IS' , 'TS', 'PAS' или 'ITA', значение **library_iim_identification.application** должно быть равно '2', '3', '4', '5', '6', '7', '10' или '11', а значение **library_iim_identification.external_file_protocols** должно соответствовать спецификациям ограничений, определенных функцией **compliant_external_file_protocol_25**.

Неформальное положение:

IP1: при ссылках на интегрированную информационную модель библиотеки LIIM 25, соответствующую настоящему стандарту для одного из классов соответствия 2, 3, 4, 5, 6 или 7, объект **library_iim_identification** может прямо или косвенно ссылаться только на файлы **http_file** типов и подтипов MIME, соответствующих общедоступ-

ным спецификациям либо спецификациям, которые можно прочитать с помощью программ, свободно предоставляемых в сети Интернет.

D.3.2 Правило allowed_entity_instance_type_in_LIIM_25_rule

Правило **allowed_entity_instance_type_in_LIIM_25_rule** устанавливает формальное ограничение на использование соответствующих допустимых типов данных **entity_instance_type**.

Для основных целей модели LIIM 25 в качестве источника данных допускают использование только следующих типов данных **entity_instance_data_type**, основанных на STEP ресурсах:

- тип данных **entity_instance_type**, ссылающийся на объект **representation**;
- тип данных **entity_instance_type**, ссылающийся на сущность **representation_context**;
- тип данных **entity_instance_type**, ссылающийся на сущность **geometric_representation_context**;
- тип данных **entity_instance_type**, ссылающийся на сущность **representation_item**;
- тип данных **entity_instance_type**, ссылающийся на сущность **date**;
- тип данных **entity_instance_type**, ссылающийся на сущность **ordinal_date**;
- тип данных **entity_instance_type**, ссылающийся на сущность **calendar_date**;
- тип данных **entity_instance_type**, ссылающийся на сущность **local_time**;
- тип данных **entity_instance_type**, ссылающийся на сущность **week_of_year_and_day_date**;
- тип данных **entity_instance_type**, ссылающийся на сущность **date_and_time**;
- тип данных **entity_instance_type**, ссылающийся на сущность **person**;
- тип данных **entity_instance_type**, ссылающийся на сущность **organization**;
- тип данных **entity_instance_type**, ссылающийся на сущность **address**.

Для основных целей модели LIIM 25 в качестве источника данных допускают использование только следующих типов данных **entity_instance_data_type**, основанных на PLIB ресурсах:

тип данных **entity_instance_type**, ссылающийся на сущность **representation_reference**;

тип данных **entity_instance_type**, ссылающийся на сущность **program_reference**;

тип данных **entity_instance_type**, ссылающийся на сущность **property_value_external_item**.

Спецификация EXPRESS:

```
*)
RULE allowed_entity_instance_type_in_LIIM_25_rule FOR (entity_instance_type);
WHERE
WR1: QUERY( x<*entity_instance_type | NOT(('REPRESENTATION_SCHEMA.REPRESENTATION'
     IN X.type_name) OR ('REPRESENTATION_SCHEMA.REPRESENTATION_CONTEXT'
     IN X.type_name) OR ('GEOMETRY_SCHEMA.GEOMETRIC_REPRESENTATION_CONTEXT'
     IN X.type_name) OR ('REPRESENTATION_SCHEMA.REPRESENTATION_ITEM'
     IN X.type_name) OR ('DATE_TIME_SCHEMA.DATE' IN X.type_name) OR
     ('DATE_TIME_SCHEMA.DATE_AND_TIME' IN X.type_name) OR
     ('DATE_TIME_SCHEMA.LOCAL_TIME' IN X.type_name) OR
     ('DATE_TIME_SCHEMA.CALENDAR_TIME' IN X.type_name) OR
     ('DATE_TIME_SCHEMA.ORDINAL_TIME' IN X.type_name) OR
     ('DATE_TIME_SCHEMA.WEEK_OF_YEAR_AND_DAY_TIME'
     IN X.type_name) OR (' PERSON_ORGANIZATION_SCHEMA.PERSON'
     IN X.type_name) OR (' PERSON_ORGANIZATION _SCHEMA.ORGANIZATION'
     IN X.type_name) OR (' PERSON_ORGANIZATION_SCHEMA.ADDRESS'
     IN X.type_name) OR ('ISO13584_EXTERNAL_FILE_SCHEMA.PROGRAM_REFERENCE'
     IN X.type_name) OR 'ISO13584_EXTERNAL_FILE_SCHEMA.REPRESENTATION_REFERENCE'
     IN X.type_name) OR ('ISO13584_EXTERNAL_FILE_SCHEMA.PROPERTY_VALUE_EXTERNAL_ITEM'
     IN X.type_name) )) = [];
END_RULE; - - allowed_entity_instance_type_in_LIIM_25_rule
(*
```

Формальное положение:

WR1: при ссылках на интегрированную информационную модель библиотеки LIIM 25, соответствующую настоящему стандарту, значение **entity_instance_type.type_name** должно ссылаться на типы данных **entity_instance_type**, допустимые в контексте обмена, установленного в настоящем стандарте.

D.3.3 Правило allowed_language_assignement_rule

Правило **allowed_language_assignement_rule** обеспечивает наличие в файле поставки библиотеки либо экземпляра **global_language_assignment**, либо одного или нескольких экземпляров **present_translation**, но не обоих экземпляров одновременно.

Спецификация EXPRESS:

```
*)
RULE allowed_language_assignment_rule FOR (present_translations,
     global_language_assignment);
WHERE
WR1: (QUERY (x <* global_language_assignment | TRUE)= [])
```

```
    XOR
        (QUERY(x<* present_translations | TRUE)=[]) ;
    END_RULE; - - Allowed_language_assignment
    (*
```

Формальное положение:

WR1: ни один из экземпляров **global_language_assignment** не доступен в файле обмена, если в нем присутствует один или несколько экземпляров **present_translation**.

D.3.4 Функция compliant_http_protocol_25

Функция **compliant_http_protocol_25** обеспечивает выполнение проверки того, допускаются или нет ссылки на протокол **external_file_protocol**, как на протокол HTTP из **library_iim_identification** со ссылками на интегрированную модель библиотеки LIIM 25 для одного из ее классов соответствия. Эта функция возвращает значение TRUE, если такие ссылки на **external_file_protocol** допускаются, и значение FALSE в противном случае. На протокол **external_file_protocol** допускаются ссылки как на протокол HTTP из **library_iim_identification**, содержащего ссылки на интегрированную модель библиотеки LIIM 25 для какого-либо ее класса соответствия, если выполняются следующие условия:

- протокол **external_file_protocol** является протоколом **http_protocol**;
- атрибут **organisation** протокола **external_file_protocol** в обязательном порядке ссылается на организацию, атрибут **id** которой равен 'IAB', а атрибут **name** — 'Internet Architecture Board';
- атрибут **protocol_name** протокола **external_file_protocol** равен 'HTTP' или 'HTTPS';
- атрибут **designation** протокола **external_file_protocol** ссылается на имена элементов **item_names**, для которых атрибут **preferred_name** равен 'Hypertext Transfer Protocol', а атрибут **short_name** имеет вид строки 'RFC' с дополнительными четырьмя цифрами и, возможно, другими символами.

Спецификация EXPRESS:

```
    *)
    FUNCTION compliant_http_protocol_25(ef : external_file_protocol):
    BOOLEAN;
    LOCAL
        ok: BOOLEAN := TRUE;
    END_LOCAL;
    IF (('ISO13584_EXTERNAL_FILE_SCHEMA' + '.HTTP_PROTOCOL' IN TYPEOF(ef)) AND
        (ef.organisation.id = 'IAB') AND (ef.organisation.name = 'Internet Architecture Board') AND
        ((ef.protocol_name = 'HTTP') OR (ef.protocol_name = 'HTTPS')) AND
        (ef.designation.preferred_name = 'Hypertext Transfer Protocol'))
    THEN
        IF 'ISO13584_IEC61360_LANGUAGE_RESOURCE_SCHEMA.TRANSLATED_LABEL'
            IN TYPEOF(ef.designation.short_name)
        THEN
            REPEAT i:= 1 TO SIZEOF(ef.designation.short_name\translated_label.labels);
                IF (ef.designation.short_name\translated_label.labels[i] LIKE 'RFC####&')
                THEN
                    ok := ok AND TRUE;
                ELSE
                    ok := OK AND FALSE;
                END_IF;
            END_REPEAT;
            RETURN(OK);
        ELSE
            IF (ef.designation.short_name LIKE 'RFC####&')
            THEN
                RETURN(TRUE);
            ELSE
                RETURN(FALSE);
            END_IF;
        END_IF;
    ELSE
        RETURN(FALSE);
    END_IF;
    END_FUNCTION; - - compliant_http_protocol_25
    (*
```

D.3.5 Функция compliant_8859_1_protocol_25

Функция **compliant_8859_1_protocol_25** обеспечивает выполнение проверки того, допускаются или нет ссылки на протокол **external_file_protocol**, как на протокол, соответствующий требованиям ИСО 8859-1, откуда

из **library_iim_identification** даются ссылки на интегрированную модель библиотеки LIIM 25 для одного из ее классов соответствия. Эта функция возвращает значение TRUE (истинно), если такие ссылки на **external_file_protocol** допускаются, и значение FALSE (ложно) в противном случае. На протокол **external_file_protocol** допускаются ссылки как на протокол ИСО 8859-1 **library_iim_identification**, откуда даются ссылки на интегрированную модель библиотеки LIIM 25 для какого-либо ее класса соответствия, если выполнены все следующие условия:

- протокол **external_file_protocol** является **standard_data_protocol**;
- атрибут **organisation** протокола **external_file_protocol** в обязательном порядке ссылается на организацию, атрибут **id** которой равен 'ISO', а атрибут **name** — 'International Organisation for Standardization';
- атрибут **protocol_name** протокола **external_file_protocol** равен 'ISO_8859_1';
- атрибут **designation** протокола **external_file_protocol** может ссылаться на наименования элементов **item_names**, для которых атрибут **preferred_name** равен 'Latin alphabet No 1', а атрибут **short_name** — 'ISO 8859-1'.

Спецификация EXPRESS:

```
*)
FUNCTION compliant_8859_1_protocol_25(ef: external_file_protocol) : BOOLEAN;
IF (('ISO13584_EXTERNAL_FILE_SCHEMA' + '.STANDARD_DATA_PROTOCOL' IN TYPEOF(ef))
      AND (ef.organisation.id = 'ISO') AND (ef.organisation.name = 'International Organisation for
      Standardization') AND (ef.protocol_name = 'ISO_8859_1') AND
      (ef.designation.preferred_name = 'Latin alphabet No 1') AND
      (ef.designation.short_name = 'ISO 8859-1'))
THEN
      RETURN(TRUE);
ELSE
      RETURN(FALSE);
END_IF;
END_FUNCTION; - - compliant_8859_1_protocol_25
(*
```

D.3.6 Функция compliant_external_file_protocol_25

Функция **compliant_external_file_protocol_25** обеспечивает выполнение проверки того, допускаются или нет ссылки на все протоколы **external_file_protocol**, как на интегрированную модель библиотеки LIIM 25, откуда из **library_iim_identification** даются ссылки на интегрированную модель библиотеки LIIM 25 для одного из ее классов соответствия 1—4. Эта функция возвращает значение TRUE, если на все протоколы **external_file_protocol** из набора **external_file_protocols** такие ссылки допускаются, и значение FALSE в противном случае.

На протокол **external_file_protocol** допускаются ссылки из объекта **library_iim_identification**, реализующего классы соответствия 1—4 интегрированной модели библиотеки LIIM 25, если на него возможны ссылки:

- как на протокол HTTP либо
- как на протокол ИСО 8859-1.

Примечание — В расширенных классах соответствия интегрированной модели библиотеки LIIM 25 допускаются ссылки на любой другой протокол **external_file_protocol**, если между отправителем и получателем заключено соответствующее соглашение.

Спецификация EXPRESS:

```
*)
FUNCTION compliant_external_file_protocol_25( s: SET [0:?] OF external_file_protocol): BOOLEAN;
REPEAT i := 1 TO SIZEOF(s);
      IF NOT (compliant_8859_1_protocol_25(s[i]) OR compliant_http_protocol_25(s[i]))
      THEN
            RETURN(FALSE);
      END_IF;
END_REPEAT;
RETURN(TRUE);
END_FUNCTION; - - compliant_external_file_protocol_25
(*
```

D.3.7 Функция is_correct_liim_25_application_value

Функция **is_correct_liim_25_application_value** обеспечивает выполнение проверки того, что объект **liim_id library_iim_identification** совместим с классом соответствия, который ассоциирован с моделью LIIM 25.

Спецификация EXPRESS:

```
*)
FUNCTION is_correct_liim_25_application_value(liim_id: library_iim_identification): BOOLEAN;
IF EXISTS(liim_id\data_exchange_specification_identification.application) AND
      (((liim_id\data_exchange_specification_identification.application[1]='2') OR
      (liim_id\data_exchange_specification_identification.application[1]='3') OR
```

```
        (liim_id\data_exchange_specification_identification.application[1]='4') OR
        (liim_id\data_exchange_specification_identification.application[1]='5') OR
        (liim_id\data_exchange_specification_identification.application[1]='6') OR
        (liim_id\data_exchange_specification_identification.application[1]='7')) AND
        (liim_id\data_exchange_specification_identification.Application LIKE '#')) OR
        ((liim_id\data_exchange_specification_identification.application[1]='1') AND
        (liim_id\data_exchange_specification_identification.application[2]='0') OR
        (liim_id\data_exchange_specification_identification.application[2]='1'))) THEN
        RETURN(TRUE);
    ELSE
        RETURN(FALSE);
    END_IF;
    END_FUNCTION; - - is_correct_liim_25_application_value
    (*
```

D.4 Дополнительное ограничение для классов соответствия 4 и 7

В данном разделе приведено дополнительное ограничение для файла поставки библиотеки, который соответствует классам соответствия 4 и 7, ассоциируемым с интегрированной информационной моделью библиотеки LIIM 25.

D.4.1 Правило nesting_level_aggregate_limit_rule

Правило **nesting_level_aggregate_limit_rule** используется для проверки того, что уровень вложения элементов в агрегированном значении не более 2.

Данное правило применяют для каждого экземпляра сущности **aggregate_type**, при этом в качестве фильтра для каждого экземпляра этой сущности используют булеву функцию **no_more_than_two_nested_levels**.

Спецификация EXPRESS:

```
    *)
    RULE nesting_level_aggregate_limit_rule FOR (library_iim_identification, aggregate_type);
        WHERE
        WR1: NOT (QUERY( liim_id <* library_iim_identification |
                (liim_id\data_exchange_specification_identification.name = 'ISO13584_25_IEC61360_5')
                AND ((liim_id\data_exchange_specification_identification.application[1]='4')
                OR (liim_id\data_exchange_specification_identification.application[1]='7'))) <> [])
              OR
                (QUERY (x <* aggregate_type | NOT no_more_than_two_nested_levels(x))= []);
    END_RULE; - - nesting_level_aggregate_limit_rule
    (*
```

D.4.2 Функция no_more_than_two_nested_levels

Функция **no_more_than_two_nested_levels** обеспечивает выполнение проверки того, что агрегированное значение не содержит более двух уровней вложения агрегированных значений.

Спецификация EXPRESS:

```
    *)
    FUNCTION no_more_than_two_nested_levels(typ : aggregate_type):BOOLEAN;
    IF NOT ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA.'+
        'ENTITY_INSTANCE_TYPE_FOR_AGGREGATE' IN TYPEOF (typ.value_type))
    THEN - - level 1 is not an aggregate
        RETURN (TRUE);
    END_IF;
    - - level 1 is an aggregate
    IF NOT ('ISO13584_IEC61360_DICTIONARY_AGGREGATE_EXTENSION_SCHEMA.'+
        'ENTITY_INSTANCE_TYPE_FOR_AGGREGATE' IN
        TYPEOF (typ.value_type.type_structure.value_type))
    THEN - - level 2 is not an aggregate
        RETURN (TRUE);
    END_IF;
    - - Level 2 is an aggregate
    RETURN(FALSE);
    END_FUNCTION; - - more_than_two_nested_levels
    (*
    *)
    END_SCHEMA; - - ISO13584_25_IEC61360_5_conformance_schema
    (*
```

Приложение Е
(обязательное)

Требования к методам реализации интегрированной информационной модели библиотеки LIIM 25

В используемых методах реализации должно быть обеспечено соответствие интегрированной информационной модели библиотеки LIIM 25. Методы реализации определяют, какие характеристики обмена необходимы для протоколов обмена.

Для файла поставки библиотеки применяют метод реализации, установленный в ИСО 10303-21.

Для внешних файлов, на которые могут быть ссылки в файле поставки библиотеки и для которых протокол **external_file_protocol** относится к стандартным данным интегрированной информационной модели библиотеки LIIM 25, методы реализации установлены в стандарте, указанном в ссылке протокола **external_file_protocol**, и при необходимости уточнены в описании стандартных данных интегрированной информационной модели библиотеки (см. приложение В).

Для структуры обмена формат файла поставки библиотеки должен соответствовать синтаксису и терминам языка EXPRESS, установленным в ИСО 10303-21 для схемы, приведенной в приложении А настоящего стандарта. В заголовке структуры обмена должно быть указано обозначение настоящего стандарта в именах схем 'ISO13584_25_IEC61360_5_library_implicit_schema'.

Примечание — Настоящий стандарт не распространяется на правила идентификации файла поставки библиотеки, которые устанавливают путем заключения соответствующего соглашения между отправителем и получателем.

Приложение F
(справочное)

Диаграммы EXPRESS-G

Рисунки F.1 и F.2 соответствуют схемам на языке EXPRESS, приведенным в разделах 6 и 7 настоящего стандарта. В этих диаграммах использованы графические обозначения EXPRESS-G для языка EXPRESS. Определение EXPRESS-G приведено в приложение А ИСО 10303-11.

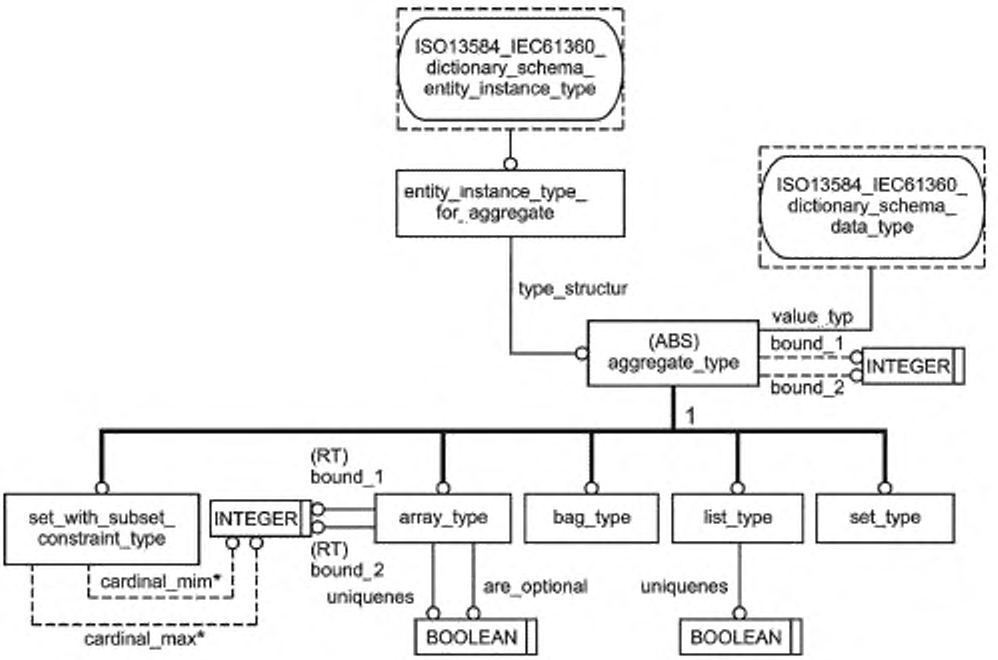


Рисунок F.1 — ISO13584_IEC61360_dictionary_aggregate_extension_schema

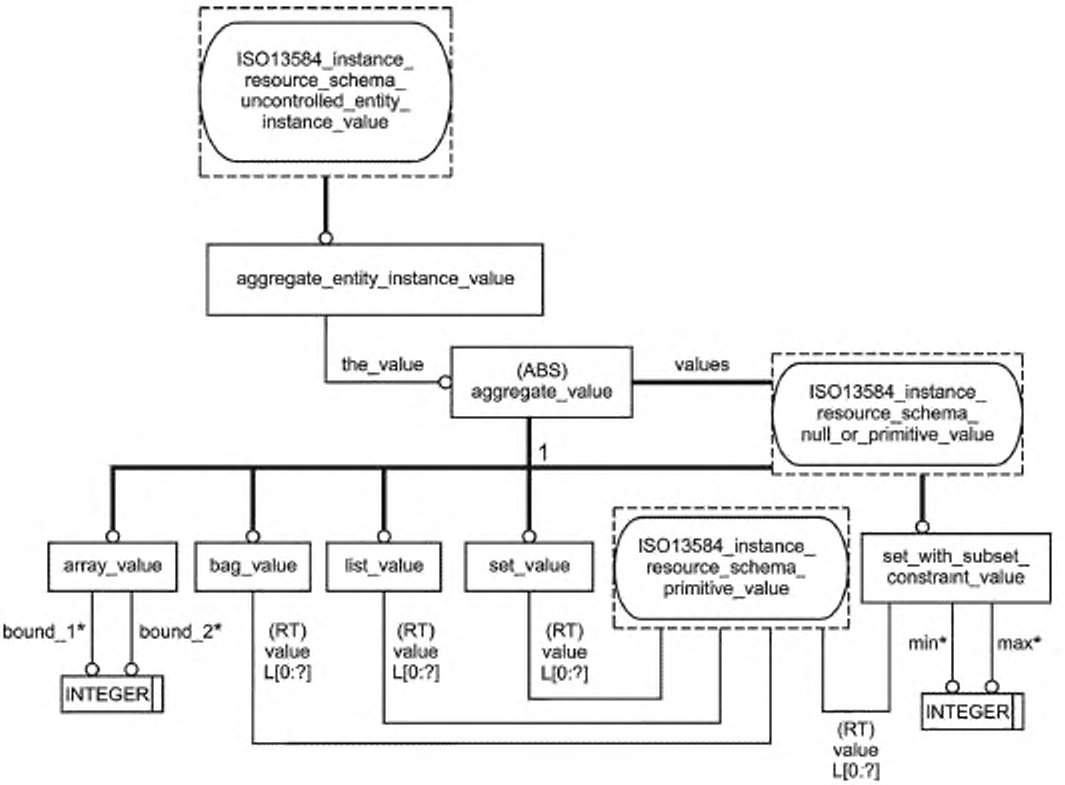


Рисунок F.2 — ISO13584_aggregate_value_schema

Приложение G
(справочное)

Примеры файлов интегрированной информационной модели библиотеки LIIM 25 с комментариями. Обмен общими моделями явного вида

В данном приложении приведены примеры разных ресурсов, входящих в описание библиотеки деталей, составленной в соответствии с требованиями настоящего стандарта.

G.1 Представление данных семейства деталей

На рисунке G.1 изображена деталь семейства, которое необходимо описать. Это семейство, обозначенное PAW, состоит из прокладочных шайб, которые предлагаются на рынке поставщиком подшипников и используются в качестве вкладышей подшипников в некоторых механических системах.

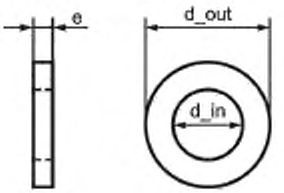


Рисунок G.1 — Вид детали семейства PAW

Описание данного семейства деталей может быть представлено на двух уровнях абстракции:

a) на уровне словаря (**ISO13584_IEC61360_dictionary_schema** и **ISO13584_extended_dictionary_schema**) приводят описания концепций семейства деталей, т. е. поставщика, класса(ов), свойства (свойств) и т. д. В этом случае определяют смысл и типы значений свойств, а также классы, в которых эти свойства используются, поставщик, установивший эти классы и т. д. Соответствующие данные элементов словарей **dictionary_element** могут быть приведены как при общем описании модели (описании состава семейства деталей), так и при ее функциональном описании (описании представления, которое может быть определено для данного семейства деталей);

b) на уровне библиотеки (**ISO13584_library_content_schema**) определяют значения свойств с целью указания допустимых для использования экземпляров семейства деталей. Данный уровень используют для описания расширения библиотеки.

Для этих двух уровней абстракции необходимы два разных способа представления.

На рисунке G.2 приведен пример экземпляра семейства, определенного только на уровне словаря. Допускается использовать только те экземпляры, значения свойств которых описывают деталь, принадлежащую типу, определенному в словаре данных.

На рисунке G.3 приведен пример семейства деталей, которое ассоциируется с описанием на двух уровнях: с использованием **dictionary_element** и библиотечной спецификации. Допускается использовать только те экземпляры, значения свойств которых описывают деталь, входящую в перечень допустимых для применения наборов значений, установленных в библиотечных спецификациях детали.

В настоящем приложении используется пример двухуровневого описания, так как данное семейство деталей изначально определено в печатном каталоге. Для вкладышей подшипников предусмотрен только строго определенный набор экземпляров.



Рисунок G.2 — Описание экземпляра из словаря

d_in	e	d_out
10	1	15
11	1	16,5
13	2	19,5
17	3	25,5
19	4	28,5

Рисунок G.3 — Пример описания семейства деталей в словаре

G.2 Описание семейства деталей PAW

Описание проводят в два этапа. Сначала определяют концепцию семейства деталей PAW и их свойства. Затем определяют допустимые экземпляры.

G.2.1 Описание словаря: механизм BSU

Для описания словаря данных, соответствующего **ISO13584_IEC61360_dictionary_schema**, необходимо определить идентификаторы (базовые семантические единицы BSU) различных понятий, используемых для определения семейства деталей. Идентификаторы предназначены для однозначного и универсального определения всех понятий словаря данных, соответствующего требованиям комплекса стандартов ИСО 13584.

В следующем примере (рисунок G.4) указаны ресурсы, которые следует использовать для определения этих идентификаторов:

/* BSU для идентификации поставщика */

/* Код для поставщика должен быть определен в соответствии с разделом «Идентификация поставщика» ИСО13584-26. Приведенный код не удовлетворяет требованиям ИСО 13584-26, так как код поставщика в данный момент неизвестен */

#20 = SUPPLIER_BSU ('INA', *);

```
/* BSU для идентификации класса component_class */
/* Единицы BSU класса определяют идентификаторы различных классов, а также поставщика, ответственного за определение класса */
#50 = CLASS_BSU ('BEARING', '001', #20);
#60 = CLASS_BSU ('PAW', '001', #20);
/* BSU для идентификации свойств */
/* Свойства BSU определяют идентификаторы свойств и класс, в котором эти свойства видимы */
#90 = PROPERTY_BSU ('d_in', '001', #50);
#100 = PROPERTY_BSU ('d_out', '001", #50);
#110 = PROPERTY_BSU ('e', '001', #60);
```

Рисунок G.4 — Идентификаторы понятий, используемых в семействе PAW

G.2.2 Описание словаря: определение элементов словаря

Единица BSU только идентифицирует понятие. Элемент **dictionary_element** приводит его определение в форме, пригодной для использования на компьютере и для чтения человеком. Связь между двумя этими уровнями описания приведена на рисунке G.5.



Рисунок G.5 — Связь между BSU и элементом словаря

На рисунке G.6 представлена основная структура элементов **dictionary_element**, соответствующих приведенным выше идентификаторам базовых семантических единиц.

```
/* Dictionary properties description */
/* supplier description */
#21 = SUPPLIER_ELEMENT(#20, $, '001', #22, #23);
#22 = ORGANIZATION($, 'INA', ' ');
#23 = ADDRESS($, $, $, $, $, $, $, 'GERMANY', $, $, $, $);
/* d_in */
#91 = NON_DEPENDENT_P_DET(#90, $, '001',
      #92, /* Item names associated to the non dependent P_DET */
      TEXT('inner diameter'), $, $, $, $, (), $, 'TO3',
      #93, /* the specific data type of the property (not represented: measure
      in mm) */
      $);
#92 = ITEM_NAMES(LABEL('inner diameter'), (), LABEL(' '), $, $);
#93 = REAL_MEASURE_TYPE('NR2..3.3', #94);
#94 = DIC_UNIT(#95, $);
#95 = SI_UNIT(*, .MILLI., .METRE.);
/* d_out */
#101 = NON_DEPENDENT_P_DET(#100, $, '001', #102, TEXT('outer
      diameter'), $, $, $, $, (), $, 'TO3', #93, $);
/* e */
#111 = NON_DEPENDENT_P_DET (#110, $, '001', #112, TEXT('thickness'),
      $, $, $, $, (), $, 'TO3', #93, $);
/* Dictionary class description */
/* Part class */
#71 = COMPONENT_CLASS (#50, /* reference to its BSU */
      $, '001',
      #72, /* item_names */
      TEXT('Class associated to the generic bearing family'),
      /* Definition */
      $, $, $, $,
      (#90, #100), /* the list of the properties that may be used
```

```
        to describe an instance of this class
        (applicability of the properties) */
        (), $, (),(), $);
#72 = ITEM_NAMES(LABEL('Generic bearing family'), (),
        LABEL('Bearing family'), $, $);
/* PAW class */
#81 = COMPONENT_CLASS(#60, $, '001', #82,
        TEXT('Class associated to the PAW part family'),
        $, $, $, #50, (#110), (), $, (),(), $);
```

Рисунок G.6 — Элемент **dictionary_element** для понятий, используемых в семействе PAW

G.2.3 Библиотечная спецификация: описание расширения класса

Расширение класса представляют в виде набора экземпляров класса. Каждый экземпляр класса содержит набор значений свойств, которые соответствуют значениям свойств, принадлежащих семейству деталей, описываемому этим экземпляром. Пример расширения класса путем представления экземпляров, приведен на рисунке G.7.

```
/* Dictionary extension */
/* Extension of a class */
#8000= EXPLICIT_ITEM_CLASS_EXTENSION(#60, /*Reference to the BSU */
        (),(),(),'001','001',(),(),(#90),
        #8100,#8200,#8300,#8400,#8500),/* the extension of class given
                                          by a list of class instances.
        .T.,$,$,(),$,(),());
```

Рисунок G.7 — Элемент **dictionary_element** для понятий, используемых в семействе PAW

Описание экземпляра класса представляют в виде списка значений свойств. На рисунке G.8 приведен пример описания путем расширения одного экземпляра, соответствующего семейству деталей, для которых значение **d_in** — 10, **e** — 1, а **d_out** — 15.

```
/* Extension of a library component */
#8100=LIB_COMPONENT_INSTANCE(#60, /* the BSU associated to the class
        which the current instance is an instance of */
        (#8101, #8102, #8103), /* the property values */
        (), $, $, $, $, .T., $);
/* Property values of the extension of a class */
#8101=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#8102=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#8103=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
```

Рисунок G.8 — Элемент **dictionary_element** для понятий, используемых в семействе PAW

G.3 Полный листинг файла для общих моделей явного вида

В данном разделе приведен пример описания существующего файла, соответствующего определениям интегрированной информационной модели библиотеки LIIM 25 для семейства деталей PAW:

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('THIS IS AN EXAMPLE OF AN EXPLICIT GENERAL MODEL'),
'2');
FILE_NAME('P25_gm_explicit.p21',
        '2000-11-28T17:38:14',
        (' '),
        ('LISI/ENSMA'),
        'ECCO RUNTIME SYSTEM BUILT-IN PREPROCESSOR V2.3beta1',
        'ECCO RUNTIME SYSTEM V2.3beta1',
        ' ');
FILE_SCHEMA (('ISO13584_25_IEC61360_5_LIBRARY_IMPLICIT_SCHEMA'));
ENDSEC;
DATA;
```

```
/* Global library description */
#2 = LIBRARY_IN_STANDARD_FORMAT ($, $, $, $, (), #20, #11, (), (),
(#20), (#50, #60), (), #3, $, $, ());
#3 = ITEM_NAMES (LABEL('Explicit general model example'), (),
LABEL(' '), $, $);
#10 = GLOBAL_LANGUAGE_ASSIGNMENT ('en');
#11 = LIBRARY_IIM_IDENTIFICATION ($, 'IS', 'ISO13584_25_IEC61360_5',
2003, '5', $, ());
/* DICTIONARY DESCRIPTION */
/*BSU for supplier */
#20 = SUPPLIER_BSU ('INA', *);
/* BSU for component_class */
#50 = CLASS_BSU ('BEARING', '001', #20);
#60 = CLASS_BSU ('PAW', '001', #20);
/* BSU for properties */
#90 = PROPERTY_BSU ('d_in', '001', #50);
#100 = PROPERTY_BSU ('d_out', '001', #50);
#110 = PROPERTY_BSU ('e', '001', #60);
/* Dictionary properties description */
/* supplier description */
#21 = SUPPLIER_ELEMENT (#20, $, '001', #22, #23);
#22 = ORGANIZATION ($, 'INA', ' ');
#23 = ADDRESS ($, $, $, $, $, $, $, 'GERMANY', $, $, $, $);
/* d_in */
#91 = NON_DEPENDENT_P_DET (#90, $, '001', #92, TEXT('inner diameter'),
$, $,
$, $, (), $, 'TO3', #93, $);
#92 = ITEM_NAMES (LABEL('inner diameter'), (), LABEL(' '), $, $);
#93 = REAL_MEASURE_TYPE ('NR2..3.3', #94);
#94 = DIC_UNIT (#95, $);
#95 = SI_UNIT (*, .MILLI., .METRE.);
/* d_out */
#101 = NON_DEPENDENT_P_DET (#100, $, '001', #102, TEXT('outer
diameter'), $,
$, $, $, (), $, 'TO3', #93, $);
#102 = ITEM_NAMES (LABEL('outer diameter'), (), LABEL(' '), $, $);
#103 = REAL_MEASURE_TYPE ('NR2..3.3', #104);
#104 = DIC_UNIT (#105, $);
#105 = SI_UNIT (*, .MILLI., .METRE.);
/* e */
#111 = NON_DEPENDENT_P_DET (#110, $, '001', #112, TEXT('thickness'), $,
$, $,
$, (), $, 'TO3', #93, $);
#112 = ITEM_NAMES (LABEL('thickness'), (), LABEL(' '), $, $);
#113 = REAL_MEASURE_TYPE ('NR2..3.3', #114);
#114 = DIC_UNIT (#115, $);
#115 = SI_UNIT (*, .MILLI., .METRE.);
/* Dictionary class description */
/* Part class */
#71 = COMPONENT_CLASS (#50, $, '001', #72, TEXT('Class associated to
the
generic bearing family'), $, $, $, $, (#90, #100), (), $, (),(), $);
#72 = ITEM_NAMES (LABEL('Generic bearing family'), (), LABEL('Bearing
family'), $,
$);
/* PAW class */
#81 = COMPONENT_CLASS (#60, $, '001', #82, TEXT('Class associated to
the PAW
part family'), $, $, $, #50, (#110), (), $, (),(), $);
#82 = ITEM_NAMES (LABEL('PAW family'), (), LABEL('PAW'), $, $);
```

```
/* Library extension */
/* Extension of a class */
#8000=
EXPLICIT_ITEM_CLASS_EXTENSION(#60,(),(),(),'001','001',(),(),(#90),
(#8100,#8200,#8300,#8400,#8500),.T.,$,$,(),$,(),());
/* Extension of a library component */
#8100=LIB_COMPONENT_INSTANCE(#60, (#8101, #8102, #8103), (), $, $, $,
$, .T., $);
/* Property values of the extension of a class */
#8101=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#8102=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#8103=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
/* Extension of a library component */
#8200=LIB_COMPONENT_INSTANCE(#60, (#8201, #8202, #8203), (), $, $, $,
$, .T., $);
/* Property values of the extension of a class */
#8201=PROPERTY_VALUE(REAL_VALUE(11.0), #90);
#8202=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#8203=PROPERTY_VALUE(REAL_VALUE(16.5), #110);
/* Extension of a library component */
#8300=LIB_COMPONENT_INSTANCE(#60, (#8301, #8302, #8303), (), $, $, $,
$, .T., $);
/* Property values of the extension of a class */
#8301=PROPERTY_VALUE(REAL_VALUE(13.0), #90);
#8302=PROPERTY_VALUE(REAL_VALUE(2.0), #100);
#8303=PROPERTY_VALUE(REAL_VALUE(19.5), #110);
/* Extension of a library component */
#8400=LIB_COMPONENT_INSTANCE(#60, (#8401, #8402, #8403), (), $, $, $,
$, .T., $);
/* Property values of the extension of a class */
#8401=PROPERTY_VALUE(REAL_VALUE(17.0), #90);
#8402=PROPERTY_VALUE(REAL_VALUE(3.0), #100);
#8403=PROPERTY_VALUE(REAL_VALUE(25.5), #110);
/* Extension of a library component */
#8500=LIB_COMPONENT_INSTANCE(#60, (#8501, #8502, #8503), (), $, $, $,
$, .T., $);
/* Property values of the extension of a class */
#8501=PROPERTY_VALUE(REAL_VALUE(19.0), #90);
#8502=PROPERTY_VALUE(REAL_VALUE(4.0), #100);
#8503=PROPERTY_VALUE(REAL_VALUE(28.5), #110);
ENDSEC;
END-ISO-10303-21;
```

Приложение Н
(справочное)

Примеры файлов интегрированной информационной модели библиотеки LIIM 25 с комментариями. Обмен функциональными моделями явного вида, соответствующими ИСО 13584-101

В данном приложении приведены примеры описания разных ресурсов, входящих в описание библиотеки деталей, составленной в соответствии с требованиями настоящего стандарта, а также пример описания файла, который может быть использован при обмене функциональными моделями явного вида, соответствующими требованиям протокола обмена видами по ИСО 13584-101.

Н.1 Описание и геометрия семейства деталей PAW

Описание семейства PAW идентично описанию с комментариями, приведенному в приложении G. Единственное отличие заключается в том, что в данное описание включены элементы поставщика, необходимые для описания вида и поставщиков геометрических характеристик.

Н.1.1 Описание словаря

Две базовые синтаксические единицы поставщика (**supplier_BSU**) LISI/ENSMA и ИСО обеспечивают идентификацию поставщиков функциональной модели и функционального вида соответственно.

```
/*BSU for supplier */
#20=SUPPLIER_BSU('94/1124946367', *); /* parts manufacturer code
following ISO 13584-26*/
#30=SUPPLIER_BSU('9/19860073600021', *);
/* LISI/ENSMA code in the coding scheme ICD=0009 : SIRET number */
#40=SUPPLIER_BSU('0112/1///13584_101_1', *);
/* Identification of ISO 13584-101 according to ISO 13584-26 */
/* supplier of the functional view */
/* BSU for component_class */
/*The two class_BSUs PAW_Gemetry and basic_geometry identify
respectively the classes corresponding to the functional model and of
the functional view. */
#50=CLASS_BSU('Bearing', '001', #20);
#60=CLASS_BSU('PAW', '001', #20);
#130=CLASS_BSU('PAW_Geometry', '001', #30);
#140=CLASS_BSU('basic_geometry', '001', #40);
/* BSU for properties */
/* The follwing property_BSU describe the properties of the part family
PAW as defined in Annex G.*/
#90=PROPERTY_BSU('d_in', '001', #50);
#100=PROPERTY_BSU('d_out', '001', #50);
#110=PROPERTY_BSU('e', '001', #50);
/* The definition of the geometry for a given part, and particularly
for the part family PAW requires the definition of representation
properties. These properties are defined as for part family properties
through the BSU mechanism. However, this BSU identifies a
representation_P_DET element.*/
#150=PROPERTY_BSU('geometry_level', '001', #140);
#160=PROPERTY_BSU('detail_level', '001', #140);
#170=PROPERTY_BSU('side', '001', #140);
#180=PROPERTY_BSU('prg', '001', #130);
#200=PROPERTY_BSU('variant', '001', #140);
#210=PROPERTY_BSU('unreg_variant', '001', #140);
/* supplier description */
/* The following supplier_element describes LISI/ENSMA as supplier. It
will be used as supplier of a functional model class. */
#31=SUPPLIER_ELEMENT(#30, $, '001', #32, #33);
#32=ORGANIZATION('LISI/ENSMA', 'LISI/ENSMA', ' ');
#33=ADDRESS($, $, $, $, $, $, $, 'FRANCE', $, $, $, $);
/* Dictionary properties description */
```

```
/* prg */
/* Data type elements associated to represetation properties are
representation_P_DETs. As example; the following data element describes
the variable allowing to refer programs (prg). Its values will be
described below. */
#91=REPRESENTATION_P_DET (#180, $, '001', #92, TEXT('variable used to
reference geometry programs'), $, $, $, $, (), $, 'A58', #93, $);
#92=ITEM_NAMES (LABEL('related program'), (), LABEL(''), $, $);
#93=PROGRAM_REFERENCE_TYPE
(('ISO13584_25_IEC61360_5_LIBRARY_IMPLICIT_SCHEMA.PROGRAM_REFERENCE'));
```

Рисунок Н.1 — Идентификаторы понятий, используемых в семействе PAW, и геометрическое представление этого семейства

Н.2 Описание геометрических представлений семейства деталей PAW

Предположим, что поставщик библиотечных данных должен определить геометрическое представление всех экземпляров семейства PAW, описанных путем расширения. Для этого необходимо представить описание класса функциональной модели.

Класс функциональной модели предназначен для представления разных видов деталей, описываемых в классе модели общего вида. Описание класса функциональной модели должно быть аналогично описанию класса модели общего вида, т. е. описание представляют путем определения класса и расширения словаря (библиотечной спецификации).

Класс **functional_model_class** описывает конкретный вид (отношение вида is-view-of) данного семейства деталей (представленного как класс модели общего вида) определенного в классе **functional_view_class**.

Например, класс **functional_view_class** для семейства деталей PAW будет определен как класс функциональной модели, представляющий некоторые геометрические аспекты.

Класс **fm_class_view_of** — это класс функциональной модели, который ссылается на точно определенный класс модели общего вида (используемый для моделирования семейства PAW) и задает конкретный тип представления (определяемый **functional_view_class**) данного класса модели общего вида.

Для класса функциональной модели не требуется представление всех значений управляющих переменных из класса функционального вида. Диапазон поддерживаемых значений указывают с помощью объектов **view_control_variable_range** согласно рисунку Н.2.

```
/* v_c_v range */
#155=VIEW_CONTROL_VARIABLE_RANGE(#150, 1, 1);
#165=VIEW_CONTROL_VARIABLE_RANGE(#160, 2, 2);
#175=VIEW_CONTROL_VARIABLE_RANGE(#170, 1, 6);
#205=VIEW_CONTROL_VARIABLE_RANGE (#200, 1, 1);
#215=VIEW_CONTROL_VARIABLE_RANGE (#210, 0, 0);
```

Рисунок Н.2 — Определение управляющих переменных для вида

В данном примере класс функциональной модели предоставляет только двумерные виды (диапазон 1...1 для #150, представляющего 'geometry_level') со стандартным представлением (диапазон 2...2 для #160, представляющего 'detail-level') всех сторон от передней ('front') до нижней ('bottom') (диапазон 1...6 для #170, представляющего 'side').

Кроме того, для создания геометрии необходимо импортировать некоторые свойства семейства PAW в класс **fm_class_view_of**. Элемент **dictionary_element** для **fm_class_view_of** представлен на рисунке Н.3.

```
/* Dictionary class description */
/* Functional model class view_of definition*/
/* The following instance describes the is_view_of relationship through
a fm_class_view_of class supplied by LISI/ENSMA.
#71=FM_CLASS_VIEW_OF(#130, /* reference to BSU */
$, '001', #72, /* item names */
TEXT('Explicit functional model class describing the 2d standard
geometry of PAW'), $, $, $, $,
(#180), /* BSU of the 'prg'and of the required_side
properties */
(), *, *, *, *, *,
#140, /* the created view (reference to the BSU of the
functional_view_class */
```

```
(#155, #165, #175, #205, #215), /* the vcv ranges */
(#150, #160, #170, #200, #210), /* vcv's imported from the
functional view class */
(), (), (), (), (), (), (), (),
#60, /* is_view_of relationship. Reference to the
functional model */
(#90, #100, #110), ),/* imported properties from the
general model */
(),(),());
#72=ITEM_NAMES(LABEL('Functional model class of PAW'), (),
LABEL('fm class of PAW'), $, $);
```

Рисунок Н.3 — Спецификация вида, создаваемого классом функциональной модели

Н.3 Библиотечная спецификация класса функциональной модели

explicit_functional_model_class_extension — это элемент **content_item,** составляющий библиотечную спецификацию подтипа **functional_model_class** и **fm_class_view_of.** Он устанавливает набор свойств, с которым должны быть сопоставлены все значения в экземпляре данного класса.

При представлении в явном виде экземплярам функциональной модели присваивают конкретные номера. В приведенном ниже экземпляре **explict_functional_model_extension** экземпляры функциональной модели описаны **lib_f_model_instance** в диапазоне от 3000 до 3450.

```
/* LIBRARY DESCRIPTION */
/* Description of the extension of a functional model class */
/* It references all the library functional model instances described
themselves by extension.
In the case of an explicit representation, the instances of a
functional model are explity enumerated. In the following
explict_functional_model_extension instance, the instances of a
functional model are described by lib_f_model_instances in the range
3000 to 3450. */
#1300=EXPLICIT_FUNCTIONAL_MODEL_CLASS_EXTENSION(#130,
(#2501, #2502, #2503, #2504, #2505, #2506), /* the set of
program references. The programs which allow to display
geometry.*/
(#7), (#12), '001', '001', (), (),
(#90, #100, #110, #170), /* properties needed to display the
geometry */
(#3000, #3010, #3020, #3030, #3040, #3050,
#3100, #3110, #3120, #3130, #3140, #3150,
#3200, #3210, #3220, #3230, #3240, #3250,
#3300, #3310, #3320, #3330, #3340, #3350,
#3400, #3410, #3420, #3430, #3440, #3450), /* The extension of
all the instances of a functional model. They are given by
the lib_f_model_instances */
.T., $, (#90, #100, #110), #180, $, $ ,(), $);
```

Рисунок Н.4 — Описание путем расширения набора экземпляров функциональной модели

/* Ссылка на программы, используемые для просмотра геометрии */

/* В приведенном выше коде использованы ссылки на программы, описывающие геометрическое представление. Ссылки program_reference относятся к внешним файлам PAW_p1.for .. PAW_p6.for, содержащим программы на языке FORTRAN, которые составлены в соответствии с протоколом обмена видами ИСО 13584-101.

```
*/
#2501=PROGRAM_REFERENCE(#7, #2601, 'Add1_PAW', 'PAW_p1',
(#90, #100, #110), (), ());
#2502=PROGRAM_REFERENCE(#7, #2602, 'Add2_PAW', 'PAW_p2',
(#90, #100, #110), (), ());
#2503=PROGRAM_REFERENCE(#7, #2603, 'Add3_PAW', 'PAW_p3',
(#90, #100, #110), (), ());
#2504=PROGRAM_REFERENCE(#7, #2604, 'Add4_PAW', 'PAW_p4',
```

```
(#90, #100, #110), (), ());
#2505=PROGRAM_REFERENCE(#7, #2605, 'Add5_PAW', 'PAW_p5',
(#90, #100, #110), (), ());
#2506=PROGRAM_REFERENCE(#7, #2606, 'Add6_PAW', 'PAW_p6',
(#90, #100, #110), (), ());
#2601=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2701));
#2602=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2702));
#2603=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2703));
#2604=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2704));
#2605=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2705));
#2606=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2706));
#2701=LANGUAGE_SPECIFIC_CONTENT((#2801), #2801, $);
#2702=LANGUAGE_SPECIFIC_CONTENT((#2802), #2802, $);
#2703=LANGUAGE_SPECIFIC_CONTENT((#2803), #2803, $);
#2704=LANGUAGE_SPECIFIC_CONTENT((#2804), #2804, $);
#2705=LANGUAGE_SPECIFIC_CONTENT((#2805), #2805, $);
#2706=LANGUAGE_SPECIFIC_CONTENT((#2806), #2806, $);
/* Descriprion of the source files for geometry */
#2801=EXTERNAL_FILE_UNIT('PAW_p1.for', '7bit');
#2802=EXTERNAL_FILE_UNIT('PAW_p2.for', '7bit');
#2803=EXTERNAL_FILE_UNIT('PAW_p3.for', '7bit');
#2804=EXTERNAL_FILE_UNIT('PAW_p4.for', '7bit');
#2805=EXTERNAL_FILE_UNIT('PAW_p5.for', '7bit');
#2806=EXTERNAL_FILE_UNIT('PAW_p6.for', '7bit');
```

Рисунок Н.5 — Ссылки на программы на языке FORTRAN для отображения геометрии

Экземпляры функциональных моделей, задаваемые путем расширения, описываются значениями своих свойств, добавленных при расширении.

На рисунке Н.6 приведен экземпляр функциональной модели, в которую путем расширения добавлены значения свойств.

```
/* Описание экземпляра функциональной модели библиотеки путем расширения */
#3000=LIB_F_MODEL_INSTANCE(#130, (#3001, #3008, #3009, #3002, #3003, #3004, #3005, #3006, #3007),
());
/* значения свойств, которые описывают расширение исходной функциональной модели библиотеки */
#3001=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#3008=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3009=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
#3002=PROPERTY_VALUE(1PROPERTY_VALUE(INTEGER_VALUE(1), #170);
#3003=PROPERTY_VALUE(1PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3004=PROPERTY_VALUE(2PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3005=PROPERTY_VALUE(1PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3006=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3007=PROPERTY_VALUE(#2501, #180);
```

Рисунок Н.6 — Связь между BSU и элементами словаря

Н.4 Полный листинг реального файла для функциональных моделей явного вида, соответствующих ИСО 13584-101

В данном разделе приведен пример реального файла. В этом файле, соответствующем интегрированной информационной модели библиотеки LIIM 25, содержится определение семейства деталей PAW и его геометрии, соответствующее протоколу обмена видами, который определен в ИСО 13584-101 и включает в себя ссылки на программы, позволяющие отображать геометрию данного семейства.

```
*/
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('PLIB EXPLICIT FUNCTIONAL MODEL EXAMPLE'), '1');
FILE_NAME('P25_fm_explicit_p101.p21',
'2001-09-21T02:38:14',
(' '),
('LISI/ENSMA'),
'ECCO RUNTIME SYSTEM BUILT-IN PREPROCESSOR V2.3.4',
```

```
'ECCO RUNTIME SYSTEM V2.3.4S',
' ');
FILE_SCHEMA(('ISO13584_25_IEC61360_5_LIBRARY_IMPLICIT_SCHEMA'));
ENDSEC;
DATA;
/* Global library description */
#2=LIBRARY_IN_STANDARD_FORMAT($, $, $, $, (), #30, #11, (#7), (#12),
(#20, #30, #40), (#50, #60, #140, #130), (), #3, $, $, ());
#3=ITEM_NAMES(LABEL('Explicit functional model: Geometry'), (),
LABEL('Geometry'), $, $);
#6=ORGANIZATION('LISI/ENSMA', 'LISI/ENSMA', ' ');
#7=STANDARD_SIMPLE_PROGRAM_PROTOCOL(#6, $, 'ISO_IS_13584_31', '001', $,
#8,$, 'FORTRAN', .SOURCE., $, $, $);
#8=ITEM_NAMES(LABEL('Geometric prog. interface'), (),
LABEL('ISO_IS_13584_31'), $, $);
#11=LIBRARY_IIM_IDENTIFICATION($, 'IS', 'ISO13584_25_IEC61360_5', 2003,
'6', $, ());
#12=VIEW_EXCHANGE_PROTOCOL_IDENTIFICATION($, 'IS', 'ISO13584_101',
2003, '2D', $, (#7), $);
#10=GLOBAL_LANGUAGE_ASSIGNMENT('en');
/* DICTIONARY DESCRIPTION */
/*BSU for suppliers */
#20=SUPPLIER_BSU('94/1124946367', *); /* parts manufacturer code
following ISO 13584-26*/
#30=SUPPLIER_BSU('9/19860073600021', *);
/* LISI/ENSMA code in the coding scheme ICD=0009 : SIRET number */
#40=SUPPLIER_BSU('0112/1///13584_101_1', *);
/* Identification of ISO 13584-101 according to ISO 13584-26 */
/* BSU for classes */
#50=CLASS_BSU('Bearing', '001', #20);
#60=CLASS_BSU('PAW', '001', #20);
#130=CLASS_BSU('PAW_Geometry', '001', #30);
#140=CLASS_BSU('basic_geometry', '001', #40);
/* BSU for properties */
#90=PROPERTY_BSU('d_in', '001', #50);
#100=PROPERTY_BSU('d_out', '001', #50);
#110=PROPERTY_BSU('e', '001', #50);
#150=PROPERTY_BSU('geometry_level', '001', #140);
#160=PROPERTY_BSU('detail_level', '001', #140);
#170=PROPERTY_BSU('side', '001', #140);
#180=PROPERTY_BSU('prg', '001', #130);
#200=PROPERTY_BSU('variant', '001', #140);
#210=PROPERTY_BSU('unreg_variant', '001', #140);
/* view control variable ranges */
#155=VIEW_CONTROL_VARIABLE_RANGE(#150, 1, 1);
#165=VIEW_CONTROL_VARIABLE_RANGE(#160, 2, 2);
#175=VIEW_CONTROL_VARIABLE_RANGE(#170, 1, 6);
#205=VIEW_CONTROL_VARIABLE_RANGE (#200, 1, 1);
#215=VIEW_CONTROL_VARIABLE_RANGE (#210, 0, 0);
/* supplier description */
#31=SUPPLIER_ELEMENT(#30, $, '001', #32, #33);
#32=ORGANIZATION('LISI/ENSMA', 'LISI/ENSMA', ' ');
#33=ADDRESS($, $, $, $, $, $, $, 'FRANCE', $, $, $, $);
/* Dictionary properties description */
/* prg */
#91=REPRESENTATION_P_DET(#180, $, '001', #92, TEXT('variable used to
reference geometry programs'), $, $, $, $, (), $, 'A58', #93, $);
#92=ITEM_NAMES(LABEL('related program'), (), LABEL(' '), $, $);
#93=PROGRAM_REFERENCE_TYPE
(('ISO13584_25_IEC61360_5_LIBRARY_IMPLICIT_SCHEMA.PROGRAM_REFERENCE'));
/* Dictionary class description */
/* Functional model class view_of definition*/
```

```
#71=FM_CLASS_VIEW_OF(#130, $, '001', #72, TEXT('Explicit functional
model class describing the 2d standard geometry of PAW'), $, $, $, $,
(#180), (), *, *, *, *, *, #140, (#155, #165, #175, #205, #215), (#150,
#160, #170, #200, #210), (), (), (), (), (), (), (), (), #60, (#90,
#100, #110), (),(),());
#72=ITEM_NAMES(LABEL('Functional model class of PAW'), (), LABEL('fm
class of PAW'), $, $);
/* LIBRARY DESCRIPTION */
#1300=EXPLICIT_FUNCTIONAL_MODEL_CLASS_EXTENSION(#130, (#2501, #2502,
#2503, #2504, #2505, #2506), (#7), (#12), '001', '001', (), (), (#90,
#100, #110, #170), (#3000, #3010, #3020, #3030, #3040, #3050, #3100,
#3110, #3120, #3130, #3140, #3150, #3200, #3210, #3220, #3230, #3240,
#3250, #3300, #3310, #3320, #3330, #3340, #3350, #3400, #3410, #3420,
#3430, #3440, #3450), .T., $, (#90, #100, #110), #180, $, $ ,(), $);
#2501=PROGRAM_REFERENCE(#7, #2601, 'Add1_PAW', 'PAW_p1', (#90, #100,
#110), (), ());
#2502=PROGRAM_REFERENCE(#7, #2602, 'Add2_PAW', 'PAW_p2', (#90, #100,
#110), (), ());
#2503=PROGRAM_REFERENCE(#7, #2603, 'Add3_PAW', 'PAW_p3', (#90, #100,
#110), (), ());
#2504=PROGRAM_REFERENCE(#7, #2604, 'Add4_PAW', 'PAW_p4', (#90, #100,
#110), (), ());
#2505=PROGRAM_REFERENCE(#7, #2605, 'Add5_PAW', 'PAW_p5', (#90, #100,
#110), (), ());
#2506=PROGRAM_REFERENCE(#7, #2606, 'Add6_PAW', 'PAW_p6', (#90, #100,
#110), (), ());
#2601=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2701));
#2602=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2702));
#2603=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2703));
#2604=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2704));
#2605=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2705));
#2606=NOT_TRANSLATABLE_EXTERNAL_CONTENT((#2706));
#2701=LANGUAGE_SPECIFIC_CONTENT((#2801), #2801, $);
#2702=LANGUAGE_SPECIFIC_CONTENT((#2802), #2802, $);
#2703=LANGUAGE_SPECIFIC_CONTENT((#2803), #2803, $);
#2704=LANGUAGE_SPECIFIC_CONTENT((#2804), #2804, $);
#2705=LANGUAGE_SPECIFIC_CONTENT((#2805), #2805, $);
#2706=LANGUAGE_SPECIFIC_CONTENT((#2806), #2806, $);
#2801=EXTERNAL_FILE_UNIT('PAW_p1.for', '7bit');
#2802=EXTERNAL_FILE_UNIT('PAW_p2.for', '7bit');
#2803=EXTERNAL_FILE_UNIT('PAW_p3.for', '7bit');
#2804=EXTERNAL_FILE_UNIT('PAW_p4.for', '7bit');
#2805=EXTERNAL_FILE_UNIT('PAW_p5.for', '7bit');
#2806=EXTERNAL_FILE_UNIT('PAW_p6.for', '7bit');
/*
GM | FV | FM
-------------------------------------------------------------------------
d_in d_out e | side | geom_level | det_level | var | unreg_var | prg
*/
#3000=LIB_F_MODEL_INSTANCE(#130, (#3001, #3008, #3009, #3002, #3003,
#3004, #3005, #3006, #3007), ());
#3001=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#3008=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3009=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
#3002=PROPERTY_VALUE(INTEGER_VALUE(1), #170);
#3003=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3004=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3005=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3006=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3007=PROPERTY_VALUE(#2501, #180);
#3010=LIB_F_MODEL_INSTANCE(#130, (#3011, #3018, #3019, #3012, #3013,
#3014, #3015, #3016, #3017), ());
```

#3011=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#3018=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3019=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
#3012=PROPERTY_VALUE(INTEGER_VALUE(2), #170);
#3013=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3014=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3015=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3016=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3017=PROPERTY_VALUE(#2502, #180);
#3020=LIB_F_MODEL_INSTANCE(#130, (#3021, #3028, #3029, #3022, #3023,
#3024, #3025, #3026, #3027), ());
#3021=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#3028=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3029=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
#3022=PROPERTY_VALUE(INTEGER_VALUE(3), #170);
#3023=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3024=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3025=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3026=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3027=PROPERTY_VALUE(#2503, #180);
#3030=LIB_F_MODEL_INSTANCE(#130, (#3031, #3038, #3039, #3032, #3033,
#3034, #3035, #3036, #3037), ());
#3031=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#3038=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3039=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
#3032=PROPERTY_VALUE(INTEGER_VALUE(4), #170);
#3033=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3034=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3035=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3036=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3037=PROPERTY_VALUE(#2504, #180);
#3040=LIB_F_MODEL_INSTANCE(#130, (#3041, #3048, #3049, #3042, #3043,
#3044, #3045, #3046, #3047), ());
#3041=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#3048=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3049=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
#3042=PROPERTY_VALUE(INTEGER_VALUE(5), #170);
#3043=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3044=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3045=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3046=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3047=PROPERTY_VALUE(#2505, #180);
#3050=LIB_F_MODEL_INSTANCE(#130, (#3051, #3058, #3059, #3052, #3053,
#3054, #3055, #3056, #3057), ());
#3051=PROPERTY_VALUE(REAL_VALUE(10.0), #90);
#3058=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3059=PROPERTY_VALUE(REAL_VALUE(15.0), #110);
#3052=PROPERTY_VALUE(INTEGER_VALUE(6), #170);
#3053=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3054=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3055=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3056=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3057=PROPERTY_VALUE(#2506, #180);
#3100=LIB_F_MODEL_INSTANCE(#130, (#3101, #3108, #3109, #3102, #3103,
#3104, #3105, #3106, #3107), ());
#3101=PROPERTY_VALUE(REAL_VALUE(11.0), #90);
#3108=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3109=PROPERTY_VALUE(REAL_VALUE(16.5), #110);
#3102=PROPERTY_VALUE(INTEGER_VALUE(1), #170);
#3103=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3104=PROPERTY_VALUE(INTEGER_VALUE(2), #160);

```
#3105=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3106=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3107=PROPERTY_VALUE(#2501, #180);
#3110=LIB_F_MODEL_INSTANCE(#130, (#3111, #3118, #3119, #3112, #3113,
#3114, #3115, #3116, #3117), ());
#3111=PROPERTY_VALUE(REAL_VALUE(11.0), #90);
#3118=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3119=PROPERTY_VALUE(REAL_VALUE(16.5), #110);
#3112=PROPERTY_VALUE(INTEGER_VALUE(2), #170);
#3113=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3114=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3115=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3116=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3117=PROPERTY_VALUE(#2502, #180);
#3120=LIB_F_MODEL_INSTANCE(#130, (#3121, #3128, #3129, #3122, #3123,
#3124, #3125, #3126, #3127), ());
#3121=PROPERTY_VALUE(REAL_VALUE(11.0), #90);
#3128=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3129=PROPERTY_VALUE(REAL_VALUE(16.5), #110);
#3122=PROPERTY_VALUE(INTEGER_VALUE(3), #170);
#3123=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3124=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3125=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3126=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3127=PROPERTY_VALUE(#2503, #180);
#3130=LIB_F_MODEL_INSTANCE(#130, (#3131, #3138, #3139, #3132, #3133,
#3134, #3135, #3136, #3137), ());
#3131=PROPERTY_VALUE(REAL_VALUE(11.0), #90);
#3138=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3139=PROPERTY_VALUE(REAL_VALUE(16.5), #110);
#3132=PROPERTY_VALUE(INTEGER_VALUE(4), #170);
#3133=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3134=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3135=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3136=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3137=PROPERTY_VALUE(#2504, #180);
#3140=LIB_F_MODEL_INSTANCE(#130, (#3141, #3148, #3149, #3142, #3143,
#3144, #3145, #3146, #3147), ());
#3141=PROPERTY_VALUE(REAL_VALUE(11.0), #90);
#3148=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3149=PROPERTY_VALUE(REAL_VALUE(16.5), #110);
#3142=PROPERTY_VALUE(INTEGER_VALUE(5), #170);
#3143=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3144=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3145=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3146=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3147=PROPERTY_VALUE(#2505, #180);
#3150=LIB_F_MODEL_INSTANCE(#130, (#3151, #3158, #3159, #3152, #3153,
#3154, #3155, #3156, #3157), ());
#3151=PROPERTY_VALUE(REAL_VALUE(11.0), #90);
#3158=PROPERTY_VALUE(REAL_VALUE(1.0), #100);
#3159=PROPERTY_VALUE(REAL_VALUE(16.5), #110);
#3152=PROPERTY_VALUE(INTEGER_VALUE(6), #170);
#3153=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3154=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3155=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3156=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3157=PROPERTY_VALUE(#2506, #180);
#3200=LIB_F_MODEL_INSTANCE(#130, (#3201, #3208, #3209, #3202, #3203,
#3204, #3205, #3206, #3207), ());
#3201=PROPERTY_VALUE(REAL_VALUE(13.0), #90);
```

#3208=PROPERTY_VALUE(REAL_VALUE(2.0), #100);
#3209=PROPERTY_VALUE(REAL_VALUE(19.5), #110);
#3202=PROPERTY_VALUE(INTEGER_VALUE(1), #170);
#3203=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3204=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3205=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3206=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3207=PROPERTY_VALUE(#2501, #180);
#3210=LIB_F_MODEL_INSTANCE(#130, (#3211, #3218, #3219, #3212, #3213,
#3214, #3215, #3216, #3217), ());
#3211=PROPERTY_VALUE(REAL_VALUE(13.0), #90);
#3218=PROPERTY_VALUE(REAL_VALUE(2.0), #100);
#3219=PROPERTY_VALUE(REAL_VALUE(19.5), #110);
#3212=PROPERTY_VALUE(INTEGER_VALUE(2), #170);
#3213=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3214=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3215=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3216=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3217=PROPERTY_VALUE(#2502, #180);
#3220=LIB_F_MODEL_INSTANCE(#130, (#3221, #3228, #3229, #3222, #3223,
#3224, #3225, #3226, #3227), ());
#3221=PROPERTY_VALUE(REAL_VALUE(13.0), #90);
#3228=PROPERTY_VALUE(REAL_VALUE(2.0), #100);
#3229=PROPERTY_VALUE(REAL_VALUE(19.5), #110);
#3222=PROPERTY_VALUE(INTEGER_VALUE(3), #170);
#3223=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3224=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3225=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3226=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3227=PROPERTY_VALUE(#2503, #180);
#3230=LIB_F_MODEL_INSTANCE(#130, (#3231, #3238, #3239, #3232, #3233,
#3234, #3235, #3236, #3237), ());
#3231=PROPERTY_VALUE(REAL_VALUE(13.0), #90);
#3238=PROPERTY_VALUE(REAL_VALUE(2.0), #100);
#3239=PROPERTY_VALUE(REAL_VALUE(19.5), #110);
#3232=PROPERTY_VALUE(INTEGER_VALUE(4), #170);
#3233=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3234=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3235=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3236=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3237=PROPERTY_VALUE(#2504, #180);
#3240=LIB_F_MODEL_INSTANCE(#130, (#3241, #3248, #3249, #3242, #3243,
#3244, #3245, #3246, #3247), ());
#3241=PROPERTY_VALUE(REAL_VALUE(13.0), #90);
#3248=PROPERTY_VALUE(REAL_VALUE(2.0), #100);
#3249=PROPERTY_VALUE(REAL_VALUE(19.5), #110);
#3242=PROPERTY_VALUE(INTEGER_VALUE(5), #170);
#3243=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3244=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3245=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3246=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3247=PROPERTY_VALUE(#2505, #180);
#3250=LIB_F_MODEL_INSTANCE(#130, (#3251, #3258, #3259, #3252, #3253,
#3254, #3255, #3256, #3257), ());
#3251=PROPERTY_VALUE(REAL_VALUE(13.0), #90);
#3258=PROPERTY_VALUE(REAL_VALUE(2.0), #100);
#3259=PROPERTY_VALUE(REAL_VALUE(19.5), #110);
#3252=PROPERTY_VALUE(INTEGER_VALUE(6), #170);
#3253=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3254=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3255=PROPERTY_VALUE(INTEGER_VALUE(1), #200);

#3256=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3257=PROPERTY_VALUE(#2506, #180);
#3300=LIB_F_MODEL_INSTANCE(#130, (#3301, #3308, #3309, #3302, #3303,
#3304, #3305, #3306, #3307), ());
#3301=PROPERTY_VALUE(REAL_VALUE(17.0), #90);
#3308=PROPERTY_VALUE(REAL_VALUE(3.0), #100);
#3309=PROPERTY_VALUE(REAL_VALUE(25.5), #110);
#3302=PROPERTY_VALUE(INTEGER_VALUE(1), #170);
#3303=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3304=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3305=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3306=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3307=PROPERTY_VALUE(#2501, #180);
#3310=LIB_F_MODEL_INSTANCE(#130, (#3311, #3318, #3319, #3312, #3313,
#3314, #3315, #3316, #3317), ());
#3311=PROPERTY_VALUE(REAL_VALUE(17.0), #90);
#3318=PROPERTY_VALUE(REAL_VALUE(3.0), #100);
#3319=PROPERTY_VALUE(REAL_VALUE(25.5), #110);
#3312=PROPERTY_VALUE(INTEGER_VALUE(2), #170);
#3313=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3314=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3315=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3316=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3317=PROPERTY_VALUE(#2502, #180);
#3320=LIB_F_MODEL_INSTANCE(#130, (#3321, #3328, #3329, #3322, #3323,
#3324, #3325, #3326, #3327), ());
#3321=PROPERTY_VALUE(REAL_VALUE(17.0), #90);
#3328=PROPERTY_VALUE(REAL_VALUE(3.0), #100);
#3329=PROPERTY_VALUE(REAL_VALUE(25.5), #110);
#3322=PROPERTY_VALUE(INTEGER_VALUE(3), #170);
#3323=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3324=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3325=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3326=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3327=PROPERTY_VALUE(#2503, #180);
#3330=LIB_F_MODEL_INSTANCE(#130, (#3331, #3338, #3339, #3332, #3333,
#3334, #3335, #3336, #3337), ());
#3331=PROPERTY_VALUE(REAL_VALUE(17.0), #90);
#3338=PROPERTY_VALUE(REAL_VALUE(3.0), #100);
#3339=PROPERTY_VALUE(REAL_VALUE(25.5), #110);
#3332=PROPERTY_VALUE(INTEGER_VALUE(4), #170);
#3333=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3334=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3335=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3336=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3337=PROPERTY_VALUE(#2504, #180);
#3340=LIB_F_MODEL_INSTANCE(#130, (#3341, #3348, #3349, #3342, #3343,
#3344, #3345, #3346, #3347), ());
#3341=PROPERTY_VALUE(REAL_VALUE(17.0), #90);
#3348=PROPERTY_VALUE(REAL_VALUE(3.0), #100);
#3349=PROPERTY_VALUE(REAL_VALUE(25.5), #110);
#3342=PROPERTY_VALUE(INTEGER_VALUE(5), #170);
#3343=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3344=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3345=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3346=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3347=PROPERTY_VALUE(#2505, #180);
#3350=LIB_F_MODEL_INSTANCE(#130, (#3351, #3358, #3359, #3352, #3353,
#3354, #3355, #3356, #3357), ());
#3351=PROPERTY_VALUE(REAL_VALUE(17.0), #90);
#3358=PROPERTY_VALUE(REAL_VALUE(3.0), #100);

#3359=PROPERTY_VALUE(REAL_VALUE(25.5), #110);
#3352=PROPERTY_VALUE(INTEGER_VALUE(6), #170);
#3353=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3354=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3355=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3356=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3357=PROPERTY_VALUE(#2506, #180);
#3400=LIB_F_MODEL_INSTANCE(#130, (#3401, #3408, #3409, #3402, #3403,
#3404, #3405, #3406, #3407), ());
#3401=PROPERTY_VALUE(REAL_VALUE(19.0), #90);
#3408=PROPERTY_VALUE(REAL_VALUE(4.0), #100);
#3409=PROPERTY_VALUE(REAL_VALUE(28.5), #110);
#3402=PROPERTY_VALUE(INTEGER_VALUE(1), #170);
#3403=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3404=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3405=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3406=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3407=PROPERTY_VALUE(#2501, #180);
#3410=LIB_F_MODEL_INSTANCE(#130, (#3411, #3418, #3419, #3412, #3413,
#3414, #3415, #3416, #3417), ());
#3411=PROPERTY_VALUE(REAL_VALUE(19.0), #90);
#3418=PROPERTY_VALUE(REAL_VALUE(4.0), #100);
#3419=PROPERTY_VALUE(REAL_VALUE(28.5), #110);
#3412=PROPERTY_VALUE(INTEGER_VALUE(2), #170);
#3413=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3414=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3415=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3416=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3417=PROPERTY_VALUE(#2502, #180);
#3420=LIB_F_MODEL_INSTANCE(#130, (#3421, #3428, #3429, #3422, #3423,
#3424, #3425, #3426, #3427), ());
#3421=PROPERTY_VALUE(REAL_VALUE(19.0), #90);
#3428=PROPERTY_VALUE(REAL_VALUE(4.0), #100);
#3429=PROPERTY_VALUE(REAL_VALUE(28.5), #110);
#3422=PROPERTY_VALUE(INTEGER_VALUE(3), #170);
#3423=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3424=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3425=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3426=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3427=PROPERTY_VALUE(#2503, #180);
#3430=LIB_F_MODEL_INSTANCE(#130, (#3431, #3438, #3439, #3432, #3433,
#3434, #3435, #3436, #3437), ());
#3431=PROPERTY_VALUE(REAL_VALUE(19.0), #90);
#3438=PROPERTY_VALUE(REAL_VALUE(4.0), #100);
#3439=PROPERTY_VALUE(REAL_VALUE(28.5), #110);
#3432=PROPERTY_VALUE(INTEGER_VALUE(4), #170);
#3433=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3434=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3435=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3436=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3437=PROPERTY_VALUE(#2504, #180);
#3440=LIB_F_MODEL_INSTANCE(#130, (#3441, #3448, #3449, #3442, #3443,
#3444, #3445, #3446, #3447), ());
#3441=PROPERTY_VALUE(REAL_VALUE(19.0), #90);
#3448=PROPERTY_VALUE(REAL_VALUE(4.0), #100);
#3449=PROPERTY_VALUE(REAL_VALUE(28.5), #110);
#3442=PROPERTY_VALUE(INTEGER_VALUE(5), #170);
#3443=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3444=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3445=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3446=PROPERTY_VALUE(INTEGER_VALUE(0), #210);

```
#3447=PROPERTY_VALUE(#2505, #180);
#3450=LIB_F_MODEL_INSTANCE(#130, (#3451, #3458, #3459, #3452, #3453,
#3454, #3455, #3456, #3457), ());
#3451=PROPERTY_VALUE(REAL_VALUE(19.0), #90);
#3458=PROPERTY_VALUE(REAL_VALUE(4.0), #100);
#3459=PROPERTY_VALUE(REAL_VALUE(28.5), #110);
#3452=PROPERTY_VALUE(INTEGER_VALUE(6), #170);
#3453=PROPERTY_VALUE(INTEGER_VALUE(1), #150);
#3454=PROPERTY_VALUE(INTEGER_VALUE(2), #160);
#3455=PROPERTY_VALUE(INTEGER_VALUE(1), #200);
#3456=PROPERTY_VALUE(INTEGER_VALUE(0), #210);
#3457=PROPERTY_VALUE(#2506, #180);
ENDSEC;
END-ISO-10303-21;
```

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов ссылочным национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандартов	Степень соответствия	Обозначение и наименование соответствующего национального стандартов
МЭК 61360-2:1998	—	*
ИСО/МЭК 8824-1	IDT	ГОСТ Р ИСО/МЭК 8824-1—2001 «Информационная технология. Абстрактная синтаксическая нотация версии один (АСН.1). Часть 1. Спецификация основной нотации»
ИСО 8859-1	—	*
ИСО 10303-11:1994	IDT	ГОСТ Р ИСО 10303-11—2009 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 11. Методы описания. Справочное руководство по языку EXPRESS»
ИСО 10303-21:2002	IDT	ГОСТ Р ИСО 10303-21—2002 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 21. Методы реализации. Кодирование открытым текстом структуры обмена»
ИСО 10303-41:2000	IDT	ГОСТ Р ИСО 10303-41—99 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 41. Интегрированные обобщенные ресурсы. Основы описания и поддержки изделий»
ИСО 10303-42:2003	—	*
ИСО 10303-43:2000	IDT	ГОСТ Р ИСО 10303-43—2002 «Системы автоматизации производства и их интеграция. Представление данных об изделии и обмен этими данными. Часть 43. Интегрированные обобщенные ресурсы. Структуры представлений»
ИСО 13584-24:2003	—	*
ИСО 13584-42:1998	—	*

* Соответствующий национальный стандарт отсутствует. 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 — идентичные стандарты.

УДК 658.52.011.56:006.354 ОКС 25.040.40 Т 58

Ключевые слова: автоматизированные промышленные системы, интеграция, жизненный цикл систем, управление производством

Редактор *Т. А. Леонова*
Технический редактор *В. Н. Прусакова*
Корректор *Р. А. Ментова*
Компьютерная верстка *Т. Ф. Кузнецовой*

Сдано в набор 03.03.2014. Подписано в печать 03.06.2014. Формат 60×84 1/8. Бумага офсетная. Гарнитура Ариал. Печать офсетная. Усл. печ. л. 8,84. Уч.-изд. л. 8,40. Тираж 65 экз. Зак. 433

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

gost.e-ecolog.ru