ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р ИСО 16100-6— 2014

Системы промышленной автоматизации и интеграция

ПРОФИЛИРОВАНИЕ ВОЗМОЖНОСТИ ИНТЕРОПЕРАБЕЛЬНОСТИ ПРОМЫШЛЕННЫХ ПРОГРАММНЫХ СРЕДСТВ

Часть 6

Службы и протоколы интерфейса для сопоставления профилей, основанных на многоцелевых структурах классов возможностей

ISO 16100-6:2011
Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 6: Interface services and protocols for matching profiles based on multiple capability class structures

(IDT))

Издание официальное

Москва
Стандартинформ
2015

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН ООО «НИИ экономики связи и информатики «Интерэкомс» (ООО «НИИ «Интерэкомс») на основе собственного аутентичного перевода на русский язык международного 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 26 ноября 2014 г. № 1871-ст

4 Настоящий стандарт идентичен международному стандарту ИСО 16100-6:2011 «Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 6. Службы и протоколы интерфейса для сопоставления профилей, основанных на многоцелевых структурах классов возможностей» (ISO 16100-6:2011 «Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 6: Interface services and protocols for matching profiles based on multiple capability class structures»).

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

5 ВВЕДЕН ВПЕРВЫЕ

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

© Стандартинформ, 2015

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

Содержание

1 Область применения 1
2 Нормативные ссылки 1
3 Термины и определения 2
4 Сокращения 3
5 Службы интерфейса поставщика сервисов 3
5.1 Наборы служб 3
5.2 Набор служб ESI 5
5.3 Служебный интерфейс импорта словаря 5
6 Расширенный служебный интерфейс 7
6.1 Службы CPTI группы 7
6.2 Расширенная CPI группа 13
6.3 Группа CCSI 20
6.4 Расширенная группа обнаружения совпадений 25
7 Формальное описание протокола расширенных сервисов ESI 29
7.1 Синтаксис основных сервисов 29
7.2 Служебный протокол CPTI группы 30
7.3 Служебные протоколы расширенной CPI группы 31
7.4 Служебные протоколы CCSI группы 33
7.5 Служебные протоколы расширенной группы обнаружения совпадений 35
8 Служба и протокол импорта словаря 35
8.1 Служба импорта словаря DictionaryImporting 35
8.2 Протокол импорта словаря DictionaryImporting 36
Приложение A (справочное) Модель возможностей, содержащая объекты данных MDD 37
Приложение B (справочное) Упрощенное сопоставление шаблонов профилей возможностей 42
Приложение C (справочное) Профили, полученные с помощью шаблона профиля возможностей 53
Приложение D (справочное) Процедура создания структуры класса возможностей 56
Приложение E (справочное) Отображение библиотеки деталей PLIB на объекты данных MDD 58
Приложение F (справочное) Отображение открытого словаря OTD на объекты данных MDD 62
Приложение G (справочное) Процедура сопоставления двух профилей 66
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации 70
Библиография 70

Введение

Разработка комплекса стандартов ИСО 16100 обусловлена необходимостью решения следующих проблем, связанных с:

a) постоянно увеличивающейся базой решений, зависящих от поставщиков;

b) трудностями, возникающими у пользователей при применении стандартов;

c) необходимостью перехода к модульным наборам инструментальных средств интеграции системы;

d) признанием того, что прикладное программное обеспечение и практический опыт его применения является интеллектуальным капиталом предприятия.

Комплекс стандартов ИСО 16100 определяет формат профиля возможностей программного обеспечения, интерпретируемого компьютером в электронно-цифровой форме и не вызывающего трудностей при его чтении человеком, а также устанавливает метод, отражающий основные возможности программного обеспечения на производстве в соответствии с ролями, определенными жизненным циклом производственного приложения, независимо от архитектуры определенной системы или платформы реализации.

Настоящий стандарт разработан Техническим комитетом ИСО/ТК 184 «Системы промышленной автоматизации и интеграция», подкомитетом ПК 5 «Архитектура, коммуникации и структуры интеграции».

Комплекс стандартов ИСО 16100 имеет общее наименование «Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств» и включает следующие части:

- часть 1. Структура;
- часть 2. Методология профилирования;
- часть 3. Службы интерфейса, протоколы и шаблоны возможностей;
- часть 4. Методы аттестационных испытаний, критерии и отчеты;
- часть 5. Методология согласования конфигураций профилей с помощью многоцелевых структур классов возможностей;
- часть 6. Службы и протоколы интерфейса для сопоставления профилей, основанных на многоцелевых структурах классов возможностей.

Некоторые из диаграмм, приведенных в настоящем стандарте, построены с использованием условных обозначений, принятых в унифицированном языке моделирования (UML). Поскольку не все принципы, используемые при построении этих диаграмм, поясняются в тексте настоящего стандарта, то предполагается, что читатель обладает определенными представлениями о языке UML.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

Системы промышленной автоматизации и интеграция

ПРОФИЛИРОВАНИЕ ВОЗМОЖНОСТИ ИНТЕРОПЕРАБЕЛЬНОСТИ ПРОМЫШЛЕННЫХ ПРОГРАММНЫХ СРЕДСТВ

Часть 6

Службы и протоколы интерфейса для сопоставления профилей, основанных на многоцелевых структурах классов возможностей

Industrial automation systems and integration. Manufacturing software capability profiling for interoperability. Part 6. Interface services and protocols for matching profiles based on multiple capability class structures

Дата введения — 2016—01—01

1 Область применения

Настоящий стандарт устанавливает службы и протоколы интерфейса, используемые для метода сопоставления, основанного на многоцелевых структурах классов возможностей. Настоящий стандарт определяет сервисные группы интерфейса шаблона профиля возможностей (CPTI), интерфейса профиля расширенных возможностей (CPI) и расширенную сервисную группу интерфейса обнаружения совпадений, которые являются расширениями сервисов Типа 1, Типа 2 и Типа 3 в соответствии с ИСО 16100-3:2005, раздел 5.4.

Настоящий стандарт определяет сервисную группу интерфейса структуры класса возможностей (CCSI) и дополнительную группу, используемую для создания, регистрации, обеспечения доступа и модификации структуры класса возможностей для ссылочных моделей производственного домена в соответствии с ИСО 16100-5:2009, раздел 6.

Настоящий стандарт устанавливает содержание особой части шаблона профиля возможностей в соответствии с ИСО 16100-5:2009, раздел 7.

2 Нормативные ссылки

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

ИСО 10303-11 Промышленные системы автоматизации и интеграция. Представление данных продукта и обмен данных. Часть 11. Метод описания: Справочное руководство по языку EXPRESS (ISO 10303-11, Industrial automation systems and integration - Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual)

ИСО 16100-1 Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 1. Структура (ISO 16100-1, Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 1: Framework)

ИСО 16100-2 Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 2. Методология профилирования (ISO 16100-2, Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 2: Profiling methodology)

ИСО 16100-3 Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 3. Службы интерфейса, протоколы и шаблоны возможностей (ISO 16100-3, Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 3: Interface services, protocols and capability templates)

Издание официальное

ИСО 16100-4 Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 4. Методы аттестационных испытаний, критерии и отчеты (ISO 16100-4, Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 4: Conformance test methods, criteria and reports)

ИСО 16100-5 Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 5. Методология профилирования сопоставления с помощью множественных структур класса возможностей (ISO 16100-5, Industrial automation systems and integration — Manufacturing software capability profiling for interoperability — Part 5: Methodology for profile matching using multiple capability class structures)

3 Термины и определения

В настоящем стандарте используются термины, определенные в ИСО 16100-1, ИСО 16100-2, ИСО 16100-3, ИСО 16100-4 и ИСО 16100-5.

3.1 **класс возможности** (capability class): Элемент метода профилирования возможности, представляющий функциональность и поведение единицы программного обеспечения в отношении программного обеспечения для производственной деятельности.

Примечание 1 — Роль блока программных средств организации производства (MSU) может быть разной в различных производственных процессах. При этом соответствующий класс возможностей MSU уникально позиционируется в структуре наследования. Он может занимать различные места в структуре агрегации.

Примечание 2 — В настоящем стандарте шаблон класса возможностей идентичен шаблону профиля возможностей (см. ИСО 16100-2:2003, 6.3, требования к шаблонам возможностей).

Примечание 3 — Адаптировано из ИСО 16100-2:2003, определение 3.3.

Примечание 4 — Класс возможностей отображается на производственный процесс. Класс возможностей существует внутри структуры наследования возможностей. Вместе с другими классами возможностей он формирует структуру агрегации возможностей.

3.2 **структура класса возможностей** (capability class structure): Иерархия класса возможностей.

Примечание — Настоящая структура предназначена для моделирования иерархий агрегации возможностей в целевых областях, определенных ИСО 16100-1:2009, рисунок 2.

3.3 **шаблон структуры класса возможностей** (capability class structure template): Схема (логическая структура в базе данных) на расширяемом языке разметки XML, представляющая собой иерархическую структуру классов возможностей.

Примечание — Адаптировано из ИСО 16100-5:2009, определение 3.2.

3.4 **шаблон профиля возможностей** (capability profile template): Схема профиля возможностей промышленных программных средств организации.

3.5 **расширенный служебный интерфейс** (extended service interface): Набор служебных точек доступа, определенных в настоящем стандарте для работы с данными производственной области, моделями производственной области, со структурой класса возможностей, профилем возможностей и шаблоном профиля возможностей.

Примечание — Термин «расширенный» относится как к сервису, описанному в настоящем стандарте, так и к «базовому» сервису, определенному в ИСО 16100-3.

3.6 **производственные данные (производственная информация)** (manufacturing domain data): Класс унифицированного языка моделирования (UML), представляющий информацию относительно производственных ресурсов, производственной деятельности или объектов, взаимодействующих в конкретной области производства.

Примечание — Адаптировано из ИСО 16100-5:2009, определение 3.3.

3.7 **шаблон производственных данных (информации)** (manufacturing domain data template): Схема (логическая структура в базе данных) на расширяемом языке разметки XML, представляющая собой производственные данные (производственную информацию).

[ИСО 16100-5:2009, определение 3.4]

3.8 **производственная модель** (manufacturing domain model): Частное представление производственного домена, состоящее из производственных данных и взаимосвязей между ними, соответствующих областям их применения.

[ИСО 16100-5:2009, определение 3.5]

3.9 **библиотека деталей** (parts library): (производственный) сборник (каталог) описаний деталей.

Примечание — Термин «библиотека деталей» также относится к словарю, например, словарю PLIB в ИСО 13584 или открытому словарю OTD в ИСО 22745.

4 Сокращения

BSU — Базовая семантическая единица (Basic Semantic Unit);

CCS — Структура класса возможностей (Capability Class Structure);

CCSI — Интерфейс структуры класса возможностей (Capability Class Structure Interface);

CPI — Интерфейс профиля возможностей (Capability Profile Interface);

CPTI — Интерфейс шаблона профиля возможностей (Capability Profile Template Interface);

CSI — Утверждение соответствия для практической реализации (Conformance Statement for the Implementation);

ESI — Интерфейс расширенных сервисов (Extended Service Interface);

ESP — Поставщик расширенных сервисов (Extended Service Provider);

ICD — Указатель международного кода (International Code Designator);

MDD — Данные производственного домена (Manufacturing Domain Data);

MDM — Модель производственного домена (Manufacturing Domain Model);

MSU — Блок программных средств организации производства (Manufacturing Software Unit);

OTD — Открытый технический словарь (Open Technical Dictionary);

PLIB — Библиотека деталей (в соответствии с ИСО 13584) (Parts Library (as specified in ISO 13584);

UML — Унифицированный язык моделирования (Unified Modeling Language);

URL — Унифицированный указатель информационного ресурса (Uniform Resource Locator);

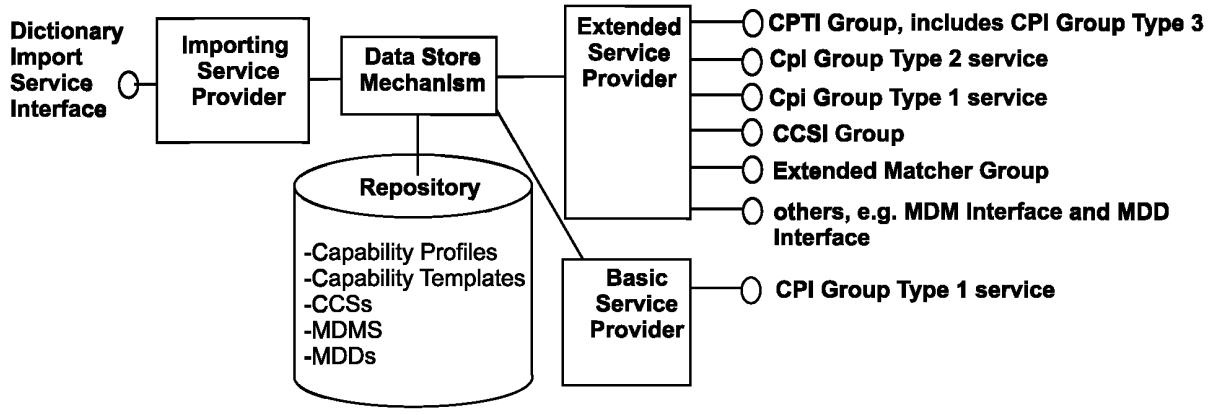
URN — Унифицированное имя ресурса (Uniform Resource Name);

XML — Расширяемый язык разметки (eXtensible Markup Language).

5 Службы интерфейса поставщика сервисов

5.1 Наборы служб

Рисунок 1 показывает все сервисы и их соотношения с Поставщиками расширенных (базовых) служб, формирующие профили возможностей, шаблоны профиля возможностей, объекты CCS, объекты модели MDM, данные MDD и объекты MDD. Поставщики базовых сервисов работают с группой услуг CPI Типа 1. Поставщики расширенных сервисов работают с услугами CPTI, услугами CPI и услугами CCSI. Кроме того, поставщики расширенных сервисов поддерживают расширенные сервисы обнаружения совпадений и другие сервисные интерфейсы, работающие с моделями MDM и объектами данных MDD.



Dictionary Import Service Interface	Сервисный интерфейс импорта словаря
Importing Service Provider	Поставщик службы импорта
Data Store Mechanism	Механизм хранения данных
Extended Service Provider	Поставщик расширенных сервисов
CPTI Group, includes CPI Group Type 3	Группа CPTI, включает группу CPI типа 3
CPI Group Type 2 service	Сервисная группа CPI типа 2
CPI Group Type 1 service	Сервисная группа CPI типа 1
CCSI Group	Группа CCSI
Extended Matcher Group	Расширенная группа обнаружения совпадений
other, e.g. MDM Interface and MDD Interface	Прочие, т.е. интерфейс модели MDM и интерфейс объекта данных MDD
Repository	Архив
Capability Profiles	Профили возможностей
Capability Templates	Шаблоны профилей возможностей
CCSs	Объекты CCS
MDMs	Объекты модели MDM
MDDs	Объекты данных MDD
Basic Service Provider	Поставщик базовых сервисов
CPI Group Type 1 service	Сервисная группа CPI типа 1

Рисунок 1 — Наборы служб поставщика расширенных сервисов

Примечание 1 — Настоящий рисунок не соответствует нотации UML. Линии, расположенные между Механизмом хранения данных и Архивом, обозначают существующие правила добавления, удаления и изменения содержания Архива. Линии, расположенные между Механизмом хранения данных и Поставщиком расширенных сервисов, обозначают отображение расширенных сервисов на сервисы хранения данных. Конкретные отображения зависят от практической реализации. В настоящем стандарте они не рассматриваются.

Примечание 2 — Элементы рисунка, выделенные жирным шрифтом, предметно рассматриваются в настоящем стандарте.

Примечание 3 — Содержание Архива хранится в виде XML-файлов.

Примечание 4 — Точка доступа ESI в настоящем стандарте представляется как объект ***ServiceAccessPoint.***

Примечание 5 — Сервисная группа CPI Типа 1, кратко описанная в ИСО 16100-3:2005, раздел 5.4, включает службу обнаружения совпадений типа 1.

Примечание 6 — Сервисная группа CPI Типа 2, кратко описаннаяв ИСО 6100-3:2005, раздел 5.4, не включает службу обнаружения совпадений типа 2, являющихся частью Расширенной группы обнаружения совпадений.

Примечание 7 — Сервисная группа CPI Типа 3 кратко описанав ИСО 16100-3:2005, раздел 5.4.

Рассматриваемые службы имеют нижеследующие характеристики:

a) для каждой службы имеется один поставщик и один пользователь: третьей стороны – нет;

b) пользователь службы может инициировать только службы, отличные от служб нижнего уровня соединений;

c) инициирование служб пользователем всегда сопровождается откликом поставщика на предоставляемую службу;

d) инициализация службы пользователем и соответствующий отклик поставщика производятся в ограниченных временных рамках в соответствии с требованиями либо пользователя службы, либо поставщика сервисовы, либо обоих сразу;

e) инициализация службы в сервисной точке доступа производится, когда закончен отклик на инициализацию предшествующей службы;

f) инициализация производится для одной отдельной службы. Инициализация не может быть выполнена для группы служб;

g) служба регистрации дополнений и обновлений в архиве использует механизм хранения данных, представленный на рисунке 1;

h) в архиве объект имеет одно из нижеследующих состояний:

1) состояние хранения: объект сохранен в Архиве после формирования запроса;

2) состояние регистрации: объект зарегистрирован в Архиве после проверки его соответствия установленным требованиям;

3) состояние стирания: объект стерт из Архива после формирования запроса на стирание.

5.2 Набор служб ESI

Основные службы ESI, обеспечиваемые ESP, могут быть взяты из группы CPI Типа 1 (см. ИСО 16100-3) и из четырех сервисных групп, перечисленных ниже и детально описанных в разделе 6. Другие сервисные группы также могут существовать (например, модели MDM, объекты MDD и объекты данных MDD), но они не рассматриваются в настоящем стандарте.

a) Сервисная группа CPTI, включающая группу CPI Типа 3, допускает:

1) создание нового шаблона профиля возможностей;

2) получение доступа к шаблону профиля возможностей;

3) модификация шаблона профиля возможностей;

4) проверка соответствия шаблона профиля возможностей;

5) регистрация профиля возможностей MSU;

6) стирание профиля возможностей MSU.

b) Сервисная группа CPI Типа 2 (см. ИСО 16100-3) допускает:

1) создание нового профиля возможностей MSU или профиля возможностей, удовлетворяющего новым требованиям;

2) обеспечение доступа к профилю возможностей MSU или к профилю возможностей, удовлетворяющему заданным требованиям;

3) модификация профиля возможностей MSU или профиля возможностей, удовлетворяющего заданным требованиям;

4) проверка соответствия профиля возможностей;

5) регистрация профиля возможностей, удовлетворяющего заданным требованиям;

6) стирание профиля возможностей, удовлетворяющего заданным требованиям.

c) Сервисная группа CCSI допускает нижеследующие услуги:

1) создание новой структуры класса возможностей;

2) обеспечение доступа к структуре класса возможностей;

3) модификация структуры класса возможностей;

4) проверка соответствия структуры класса возможностей;

5) регистрация структуры класса возможностей;

6) стирание структуры класса возможностей.

d) Расширенная группа обнаружения совпадений допускает:

1) обеспечение доступа к профилю возможностей MSU или профилю возможностей, удовлетворяющему заданным требованиям;

2) сопоставление двух профилей возможностей, при этом каждый ссылается на другую структуру класса возможностей.

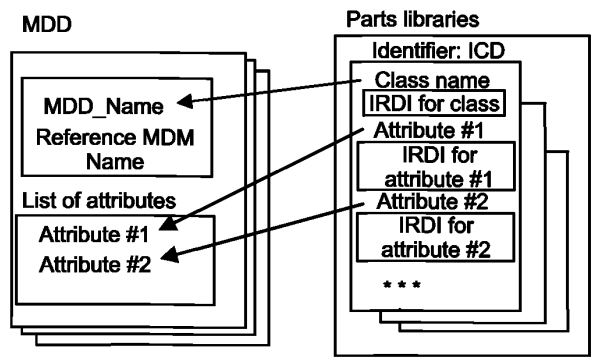
5.3 Служебный интерфейс импорта словаря

5.3.1 Библиотеки деталей, импортированные в Архив

Служебный интерфейс импорта словаря обеспечивает импорт словаря ((например, словаря деталей PLIB или открытого словаря OTD) в архив.

5.3.2 Соотношение между библиотеками деталей и объектами MDD

Библиотека деталей, импортированная в Архив, может быть использована как часть объектов MDD при профилировании возможностей. Объекты MDD в модели MDM включают определение производственного процесса. Словарь деталей PLIB и открытый технический словарь OTD включают определения технических терминов: это может быть использовано для профилирования возможностей по аналогии с использованием объектов MDD. Все классы и атрибуты словаря PLIB и словаря OTD могут быть идентифицированы уникальным идентификатором в соответствии с ИСО 29002-5 и ИСО 22745-13. Уникальный идентификатор – это форма международной регистрации идентификатора данных (IRDI) в соответствии с ИСО/МЭК 11179-5. Классы и атрибуты словаря PLIB могут также быть описаны комбинацией некоторого словаря и базовой семантической единице BSU, кодом рассматриваемого класса PLIB и внутренним атрибутом словаря. Базовая семантическая единица BSU, соответствующая атрибуту словаря PLIB – это комбинация кода атрибута и BSU родительского класса. Для словаря OTD нет необходимости ссылаться на BSU родительского класса, так как ИСО 22745 нейтрален по отношению к классификации. Он определяет свойства в OTD независимо от класса. Соотношение между словарем PLIB, словарем OTD и объектом MDD определяется отображением уникального идентификатора на каждый объект MDD и атрибут MDD как показано на рисунке 2.



MDD	Объект MDD
MDD_Name	Имя объекта MDD
Reference MDM Name	Ссылочное имя модели MDM
List of attributes	Список атрибутов
Attribute #1	Атрибут 1
Parts libraries	Библиотека деталей
Identifier: ICD	Идентификатор ICD
Class name	Имя класса
IRDI for class	Международный зарегистрированный идентификатор данных для класса
IRDI for attribute #1	Международный зарегистрированный идентификатор данных для атрибута 1

Примечание 1 — Настоящий рисунок не соответствует нотации UML.

Примечание 2 — В ИСО 13584 термин «свойство» используется вместо термина «атрибут».

Рисунок 2 — Соотношения между библиотеками деталей и объектами MDD

5.3.3 Отображение элемента PLIB на объект MDD

Элемент «**MDD_name**» на рисунке 2 должен иметь нижеследующие расширенные атрибуты:
- «**dictionary_id**» (идентификатор словаря),
- «**parent**» (родитель),
- «**BSU**» (Базовая семантическая единица),
- «**version**» (версия)
- «**revision**» (пересмотр).

Набор атрибутов «**dictionary_id**», «**parent**» и «BSU» может быть использован как идентификатор объекта MDD.

В соответствии с рисунком 2, атрибут объекта MDD относится к атрибутам продуктов словаря PLIB. Объект MDD соответствует элементу словаря PLIB. Атрибуты, принадлежащие одному элементу, ассоциируются с одним объектом MDD. См. приложение E.

6 Расширенный служебный интерфейс

6.1 Службы CPTI группы

6.1.1 Сценарии, используемые группой услуг CPTI

Группа CPTI использует нижеследующие сценарии шаблона профиля возможностей:

a) создать шаблон профиля возможностей для особой структуры класса либо путем частичного заполнения характеристической формальной структуры шаблона профиля возможностей, либо путем модификации существующего шаблона профиля возможностей с новым идентификатором ID шаблона профиля возможностей, используя объекты MDD в модели MDM, а затем получить шаблон профиля возможностей от поставщика сервисов;

b) запросить шаблон профиля возможностей из архива с помощью идентификатора ID шаблона профиля возможностей, а затем получить шаблон профиля возможностей от поставщика сервисов;

c) модифицировать шаблон профиля возможностей в соответствии либо с требованиями пользователя, либо с результатами проверки соответствия, а затем получить модифицированный шаблон профиля возможностей от поставщика сервисов;

d) испытать шаблон профиля возможностей в сравнении с критериями соответствия шаблона профиля возможностей, данными в ИСО 16100-5:2009, раздел 8, а затем получить положительный отклик, отрицательный отклик или отклик уровня сопоставления от поставщика сервисов;

e) зарегистрировать испытанный шаблон профиля возможностей в архиве и получить статус «зарегистрирован» от поставщика сервисов;

f) стереть шаблон профиля возможностей, основанный на идентификаторе шаблона профиля возможностей, и получить статус “стерт” от поставщика сервисов.

6.1.2 Создание шаблона профиля возможностей

6.1.2.1 Процесс создания

Процесс создания шаблона профиля возможностей для класса возможностей рассматриваемой структуры класса возможностей состоит из конфигурирования характеристической формальной структуры шаблона профиля возможностей путем:

a) внесения особых значений определенных атрибутов определенных элементов в характеристическую формальную структуру шаблона профиля возможностей в соответствии с ИСО 16100-5:2009, раздел 6.3;

b) модификации ранее внесенных значений в существующем шаблоне профиля возможностей.

6.1.2.2 Конфигурация шаблона профиля возможностей

Характеристическая формальная структура шаблона профиля возможностей может быть либо частично, либо полностью заполнена в соответствии с требованиями приложения.

Для любого шаблона профиля возможностей должны быть внесены или модифицированы нижеследующие атрибуты и элементы:

a) атрибуты «идентификатор» и «название» в элементе «Шаблон профиля возможностей»;

b) атрибут имени области «**domain_name**» в элементе ссылочного имени «**Reference_MDM_name**»;

c) атрибут названия формата «**format_name**» в элементе описания формата «**MDD_Description_format**» с одним из четырех нижеследующих значений:

1) «**Set_Of_MDD_objects**» (набор объектов MDD);

2) «**List_Of_MDD_objects**» (список объектов MDD);

3) «**Time_Ordered_MDD_objects**» (объекты MDD, упорядоченные по времени);

4) «**Event_Ordered_MDD_objects**» (объекты MDD в порядке поступления);

d) атрибуты «название» и «производственная деятельность» в элементе названия объекта «**MDD_name**».

Каждое значение, внесенное или модифицированное для атрибутов «идентификатор» (в разделе 6.1.2.2 a), имени области «**domain_name**» в разделе 6.1.2.2 b) и «название» в разделе 6.1.2.2 d), должно быть уникальным.

Шаблон профиля возможностей должен быть использован для создания профиля возможностей, ассоциированного с классом возможностей.

6.1.2.3 Служба *createTemplate*

6.1.2.3.1 Шаблон, основанный на характеристической структуре формальной возможности

Служба createTemplate должна позволять пользователю шаблона создавать шаблон, основанный на характеристической структуре формальной возможности. Если создание шаблона основано на структуре формальной возможности, то служба ***createTemplate*** использует, как минимум, ***requestBlankTemplate, returnBlankTemplate, ProcessFilledTemplate*** и службу returnProcessingResult. Служба ***createTemplate*** включает нижеследующие шаги:

a) пользователь шаблона инициирует службу ***requestBlankTemplate*** объекта ***ServiceAccessPoint***, в котором нет параметров, ассоциированных с услугой ***requestBlankTemplate;***

b) поставщик сервисов инициирует службу ***returnBlankTemplate*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnBlankTemplate*** являются заготовка шаблона и ошибка создания;

c) пользователь шаблона заполняет заготовку шаблона, используя объекты MDD модели MDM, а затем инициирует службу ***processFilledTemplate*** объекта ***ServiceAccessPoint***, в котором параметром ***processFilledTemplate*** является идентификатор шаблона;

d) поставщик сервисов проверяет уникальность идентификатора шаблона, а затем инициирует службу ***returnProcessingResult*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 3 (выше пунктирной линии) на языке UML приведена диаграмма последовательности, являющаяся обязательным шагом процедуры создания шаблона из структуры формального шаблона.

6.1.2.3.2 Шаблон, созданный путем модификации существующего шаблона профиля возможностей

Служба ***createTemplate*** дает возможность пользователю шаблона создать новый шаблон путем модификации существующего шаблона. Если при создании нового шаблона модифицируется существующий шаблон, то служба ***createTemplate*** использует, как минимум, службу ***requestExistingTemplate,*** службу ***returnExistingTemplate,*** службу ***processModifiedTemplate*** и службу ***returnProcessingResult***. Служба ***createTemplate*** включает нижеследующие шаги:

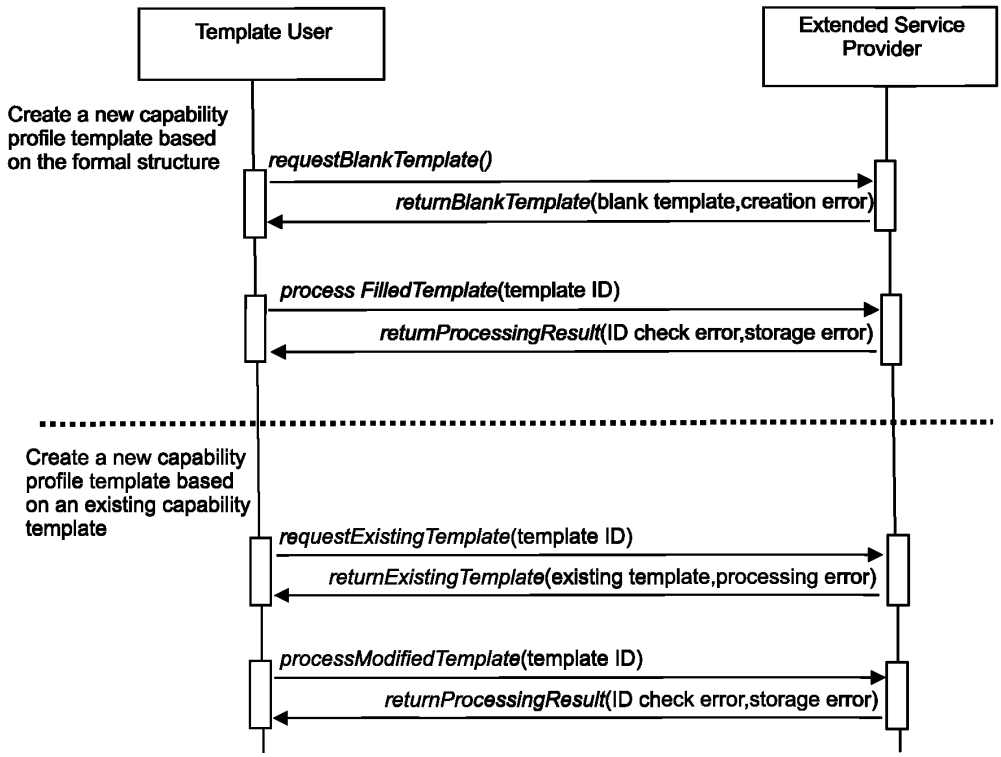
a) пользователь шаблона инициирует службу ***requestExistingTemplate*** объекта ***ServiceAccessPoint,*** в котором параметром службы ***requestExistingTemplate*** является идентификатор шаблона;

b) поставщик сервисов инициирует службу ***returnExistingTemplate*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***requestExistingTemplate*** являются существующий шаблон и ошибка обработки;

c) пользователь шаблона модифицирует существующий шаблон, а затем инициирует службу ***processModifiedTemplate*** объекта ***ServiceAccessPoint***, в котором параметром службы ***processModifiedTemplate*** является идентификатор шаблона;

d) поставщик сервисов проверяет уникальность идентификатора шаблона, а затем инициирует службу returnProcessingResult объекта ***ServiceAccessPoint***, в котором параметрами услуги ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 3 (ниже пунктирной линии) на языке UML показана диаграмма последовательности, являющаяся обязательным шагом создания шаблона путем модификации существующего шаблона.



Template User	Пользователь шаблона
Extended Service Provider	Поставщик расширенных сервисов
Create a new capability profile template based on the formal structure	Создание нового шаблона профиля возможностей, основанного на использовании формальной структуры
requestBlankTemplate()	***requestBlankTemplate*** ()
returnBlankTemplate(blank template, creation error	***returnBlankTemplate*** (заготовка шаблона, ошибка создания)
processFilledTemplate(template ID)	***processFilledTemplate*** (идентификатор шаблона)
returnProcessingResult(ID check error, storage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)
Create a new capability profile template based on an existing capability template	Создание нового шаблона профиля возможностей на основе использования существующего шаблона профиля возможностей
requestExistingTemplate(template ID)	***requestExistingTemplate*** (идентификатор шаблона)
returnExistingTemplate(existing template, processing error)	***ReturnExistingTemplate*** (существующий шаблон, ошибка обработки)
processModifiedTemplate (template ID)	***processModifiedTemplate*** (идентификатор шаблона)
returnProcessingResult (ID check error, storage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)

Примечание 1 — Пунктирная линия разделяет два рассмотренных варианта.

Примечание 2 — Созданию шаблона всегда предшествуют предварительные шаги регистрации и верификации уникального идентификатора шаблона, лежащие вне области применения ИСО 16100.

Рисунок 3 — Служба ***createTemplate***

6.1.3 Служба ***accessTemplate***

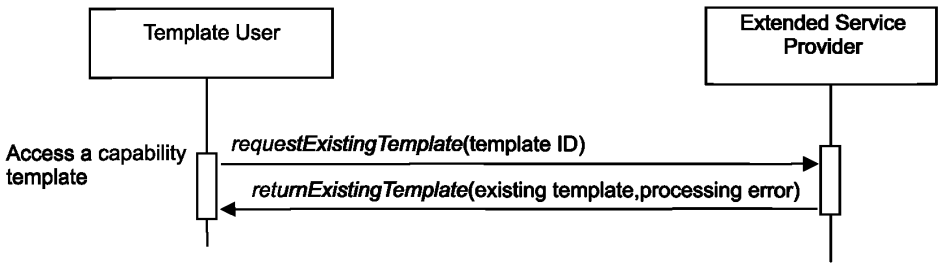
Служба ***accessTemplate,*** использующая службу ***requestExistingTemplate*** и службу ***returnExistingTemplate,*** дает возможность пользователю шаблона получить доступ к какому-либо другому существующему шаблону.

Служба ***accessTemplate*** включает нижеследующие шаги:

а) пользователь шаблона инициирует службу ***requestExistingTemplate*** объекта ***ServiceAccessPoint***, в котором параметром службу requestExistingTemplate является идентификатор шаблона;

b) поставщик сервисов инициирует службу ***returnExistingTemplate*** объекта ***ServiceAccessPoint***, в котором параметрами службы requestExistingTemplate являются существующий шаблон и ошибка обработки.

На рисунке 4 на языке UML показана диаграмма последовательности обязательных шагов процедуры получения доступа к шаблону, основанной на использовании идентификатора шаблона.



Template User	Пользователь шаблона
Extended Service Provider	Поставщик расширенных сервисов
Access a capability template	Получение доступа к шаблону возможностей
requestExistingTemplate (template ID)	***requestExistingTemplate*** (идентификатор шаблона)
returnExistingTemplate (existing template, processing error)	***returnExistingTemplate*** (существующий шаблон, ошибка обработки)

Рисунок 4 — Служба accessTemplate

6.1.4 Служба *modifyTemplate*

Служба ***modifyTemplate***, использующая службу ***requestExistingTemplate***, службу ***returnExistingTemplate,*** службу ***processModifiedTemplate*** и службу ***returnProcessingResult***, дает возможность пользователю шаблона модифицировать существующий шаблон.

Служба ***modifyTemplate*** включает нижеследующие шаги:

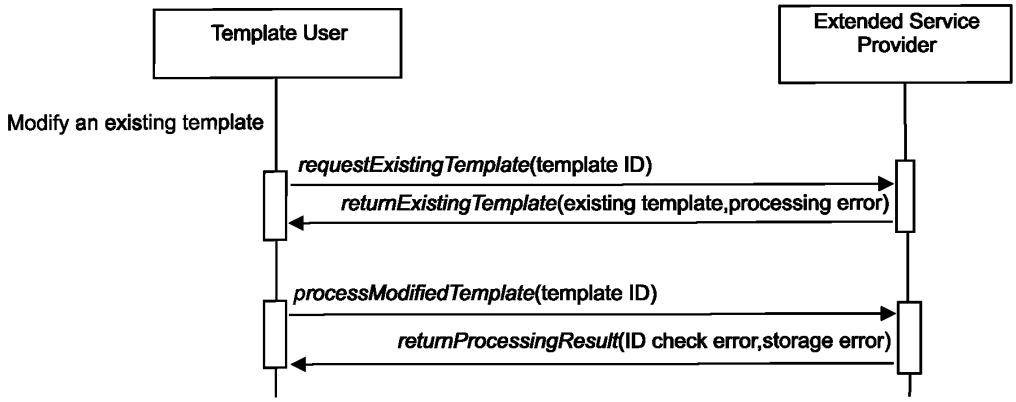
a) пользователь шаблона инициирует службу requestExistingTemplate объекта ***ServiceAccessPoint***, в котором параметром службу ***requestExistingTemplate*** является идентификатор шаблона;

b) поставщик сервисов инициирует службу ***returnExistingTemplate*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***requestExistingTemplate*** являются существующий шаблон и ошибка обработки;

c) пользователь шаблона модифицирует существующий шаблон, а затем инициирует службу ***processModifiedTemplate*** объекта ***ServiceAccessPoint***, в котором параметром службы ***processModifiedTemplate*** является идентификатор шаблона;

d) поставщик сервисов проверяет уникальность идентификатора шаблона, а затем инициирует службу ***returnProcessingResult*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 5 на языке UML приведена диаграмма обязательных шагов последовательности процедуры модификации существующего шаблона с помощью идентификатора шаблона.



Template User	Пользователь шаблона
Extended Service Provider	Поставщик расширенных сервисов
Modify an existing template	Модификация существующего шаблона
requestExistingTemplate (template ID)	***requestExistingTemplate*** (идентификатор шаблона)
returnExistingTemplate (existing template, processing error)	***returnExistingTemplate*** (существующий шаблон, ошибка обработки)
processModifiedTemplate (template ID)	***processModifiedTemplate*** (идентификатор шаблона)
returnProcessingResult (ID checke error, strorage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)

Рисунок 5 — Служба ***modifyTemplate***

6.1.5 Служба *validateTemplate*

Служба ***validateTemplate,*** использующая службу ***requestUnregisteredTemplate,*** службу ***returnUnregisteredTemplate,*** службу ***testTemplate,*** службу ***returnTestResult***, дает возможность пользователю шаблона провести проверку соответствия существующего незарегистрированного шаблона.

Служба ***validateTemplate*** включает нижеследующие шаги:

a) пользователь шаблона инициирует службу ***requestUnregisteredTemplate*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestUnregisteredTemplate*** является идентификатор шаблона;

b) поставщик сервисов инициирует службу ***returnUnregisteredTemplate*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnUnregisteredTemplate*** являются незарегистрированный шаблон и ошибка обработки;

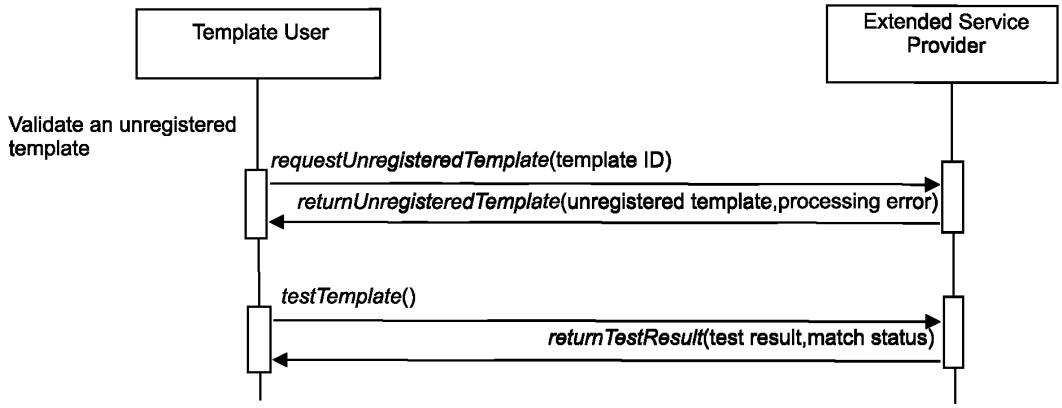
c) пользователь шаблона инициирует службу ***testTemplate*** объекта ***ServiceAccessPoint***, в котором нет параметров, ассоциированных со службой ***testTemplate***;

d) поставщик сервисов инициирует службу ***returnTestResult*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnTestResult*** являются результат испытаний и статус сопоставления.

Значение параметра результата испытаний службы ***returnTestResult*** может быть положительным, отрицательным или уровнем сопоставления, определенным в ИСО 16100-5:2009, раздел 7.2. Значение параметра статуса сопоставления службы ***returnTestResult*** должно быть эквивалентно выходным сообщениям, рассмотренным в ИСО 16100-4:2006, раздел В.1.

Испытанный шаблон должен быть зарегистрирован в архиве, если значение параметра результата испытаний службы ***returnTestResult*** «положительно» или «полное совпадение» (см. ИСО 16100-5:2009, раздел 7.2). В зависимости от требований пользователя, испытанный шаблон может быть зарегистрирован в архиве, если значение параметра результата испытаний либо «полное обязательное совпадение», либо «некоторое обязательное совпадение» (см. ИСО 16100-5:2009, раздел 7.2). Испытанный шаблон может быть модифицирован снова, чтобы удовлетворять критериям соответствия, если значение параметра результата испытаний отрицательно или «обязательное совпадение отсутствует» (см. ИСО 16100-5:2009, раздел 7.2).

На рисунке 6 на языке UML приведена диаграмма обязательных шагов последовательности проверки соответствия незарегистрированного шаблона, основанной на использовании идентификатора шаблона.



Template User	Пользователь шаблона
Extended Service Provider	Поставщик расширенных сервисов
Validate an unregistered template	Валидация незарегистрированного шаблона
requestUnregisteredTemplate (template ID)	***requestUnregisteredTemplate*** (идентификатор шаблона)
returnUnregisteredTemplate (unregistered template, processing error)	***returnUnregisteredTemplate*** (незарегистрированный шаблон, ошибка обработки)
testTemplate	***testTemplate***()
returnTestResult(test result, match status)	***returnTestResult*** (результат испытаний, статус сопоставления)

Рисунок 6 — Служба ***validateTemplate***

6.1.6 Служба *deleteTemplate*

Служба ***deleteTemplate,*** использующая службу ***requestExistingTemplate,*** службу ***returnExistingTemplate,*** службу ***removeTemplate*** и службу ***returnRemoveResult***, дает возможность пользователю шаблона стереть существующий шаблон.

Служба ***deleteTemplate*** включает нижеследующие шаги:

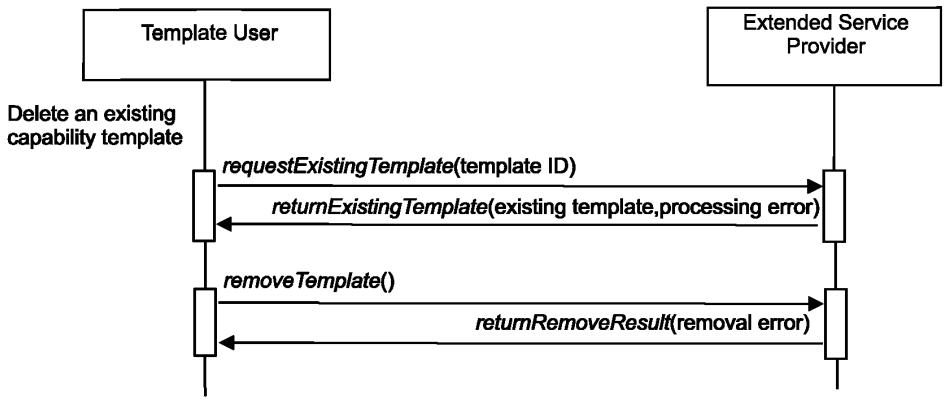
a) пользователь шаблона инициирует службу ***requestExistingTemplate*** объекта ***ServiceAccessPoint***, в котором параметром службу ***requestExistingTemplate*** является идентификатор шаблона;

b) поставщик сервисов инициирует службу ***returnExistingTemplate*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***requestExistingTemplate*** являются существующий шаблон и ошибка обработки;

c) пользователь шаблона инициирует службу ***removeTemplate*** объекта ***ServiceAccessPoint***, в котором отсутствуют параметры, ассоциированные со службой ***removeTemplate*;**

d) поставщик сервисов инициирует службы ***returnRemoveResult*** объекта ***ServiceAccessPoint***, в котором параметром службы ***returnRemoveResult*** является ошибка удаления.

На рисунке 7 на языке UML приведена диаграмма обязательных шагов последовательности стирания шаблона, основанная на использовании идентификатор шаблона.



Template User	Пользователь шаблона
Extended Service Provider	Поставщик расширенных сервисов
Delete an existing capability template	Стирание существующего шаблона возможностей
requestExistingTemplate (template ID)	***requestExistingTemplate*** (идентификатор шаблона)
returnExistingTemplate (existing template, processing error)	***returnExistingTemplate*** (существующий шаблон, ошибка обработки)
removeTemplate ()	***removeTemplate***()
returnRemoveResult (removal error)	***returnRemoveResult*** (ошибка удаления)

Рисунок 7 — Служба ***deleteTemplate***

6.2 Расширенная CPI группа

6.2.1 Сценарии, используемые расширенной CPI группой сервисов

Группа сервисов CPI использует нижеследующий сценарий профиля возможностей:

a) создать профиль возможностей либо путем внесения, по крайней мере, необходимых атрибутов (элементов) шаблона, либо путем модификации существующего профиля возможностей с новым идентификатором профиля, используя объекты MDD модели MDM, а затем получения профиля возможностей от поставщика сервисов;

b) получение доступа к профилю либо из Архива через интерфейс ESI, либо из MSU, путем использования идентификатора профиля возможностей, а затем получение профиля возможностей либо от поставщика сервисов, либо из MSU;

c) модификация профиля возможностей в соответствии либо с требованиями пользователя, либо в соответствии с результатами проверки соответствия, а затем получение модифицированного профиля возможностей либо от поставщика сервисов, либо из MSU;

d) испытание профиля возможностей по его критериям соответствия и получение либо положительного отклика, либо отрицательного отклика, либо отклика уровня сопоставления от поставщика сервисов;

e) регистрация испытанного профиля возможностей в Архиве и получение статуса «зарегистрирован» от поставщика сервисов;

f) стирание профиля возможностей с помощью идентификатора профиля возможностей и получение статуса «стерт» от поставщика сервисов.

6.2.2 Создание профиля возможностей

6.2.2.1 Процесс создания

Создание профиля возможностей производится либо путем внесения особых значений определенных атрибутов определенных элементов в шаблон профиля возможностей, либо путем модификации ранее внесенных значений в существующий профиль возможностей.

6.2.2.2 Служба ***createProfile***

6.2.2.2.1 Профиль, основанный на шаблоне профиля возможностей

Служба ***createProfile*** дает возможность пользователю профиля запросить создание профиля, основанного на использовании шаблона профиля возможностей. Если создание профиля основано

на использовании шаблона профиля возможностей, то служба ***createProfile*** использует, как минимум, службу ***requestExistingTemplate***, службу ***returnExistingTemplate***, службу ***processFilledProfile*** и службу ***returnProcessingResult***. Служба ***createProfile*** включает нижеследующие шаги:

a) пользователь профиля инициирует службу ***requestExistingTemplate*** объекта ***ServiceAccessPoint***, в котором параметром службу ***requestExistingTemplate*** является идентификатор шаблона;

b) поставщик сервисов инициирует службу ***returnExistingTemplate*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***requestExistingTemplate*** являются существующий шаблон и ошибка обработки;

c) пользователь профиля заполняет существующий шаблон, используя объект MDD модели MDM, а затем инициирует службу ***processFilledProfile*** объекта ***ServiceAccessPoint***, в котором параметром службы ***processFilledProfile*** является идентификатор профиля;

d) поставщик сервисов проверяет уникальность идентификатора профиля, а затем инициирует службу ***returnProcessingResult*** объекта ***ServiceAccessPoint***, в котором параметрами услуги ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 8 (выше пунктирной линии) на языке UML приведена диаграмма последовательности обязательных шагов создания профиля с помощью существующего шаблона.

6.2.2.2.2 Профиль, основанный на существующем профиле возможностей

Служба ***createProfile*** дает возможность пользователю профиля запросить создание нового профиля путем модификации существующего профиля. При создании нового профиля путем модификации существующего профиля, служба ***createTemplate*** использует, как минимум, службу ***requestExistingProfile***, службу ***returnExistingProfile***, службу ***processModifiedProfile*** и службу ***returnProcessingResult***. Служба ***createProfile*** включает нижеследующие шаги:

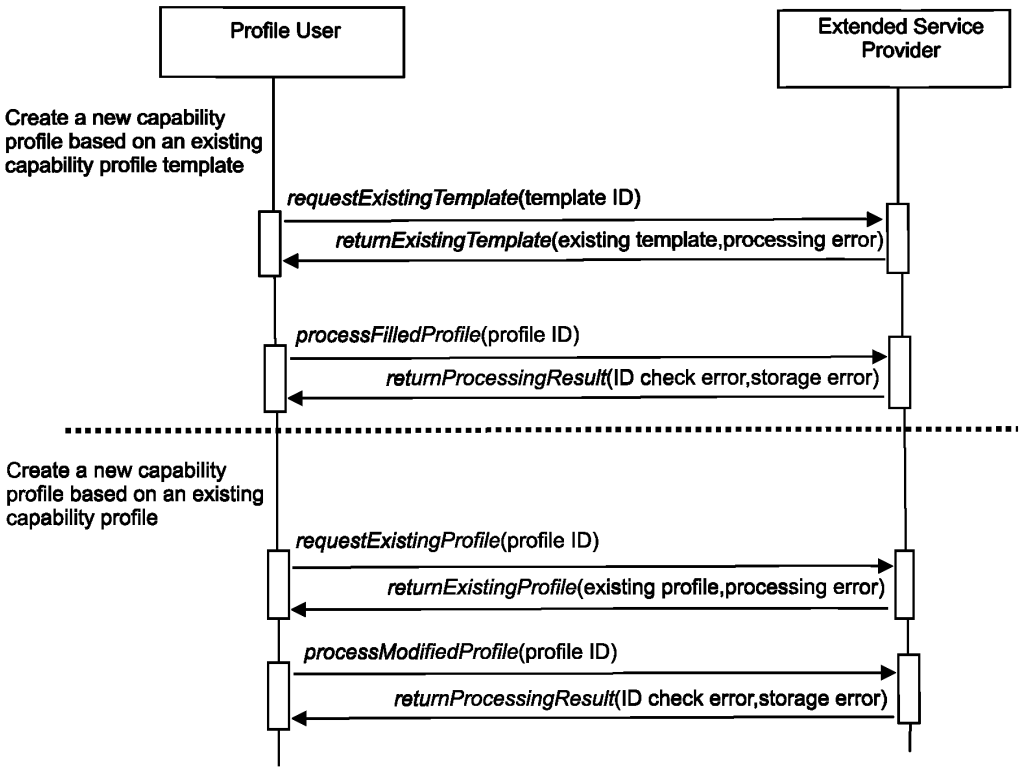
a) пользователь профиля инициирует службу ***requestExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingProfile*** является идентификатор профиля;

b) поставщик сервисов инициирует службу ***returnExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingProfile*** являются существующий профиль и ошибка обработки;

c) пользователь профиля модифицирует существующий профиль, используя объект данных MDD модели MDM, а затем инициирует службу ***processModifiedProfile*** объекта ***ServiceAccessPoint***, в котором параметром службы ***processModifiedProfile*** является идентификатор профиля;

d) поставщик сервисов проверяет уникальность идентификатора профиля, а затем инициирует службу ***returnProcessingResult*** объекта ***ServiceAccessPoint***, в котором параметрами услуги ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 8 (ниже пунктирной линии) на языке UML приведена диаграмма последовательности создания нового профиля из существующего профиля.



Template User	Пользователь профиля
Extended Service Provider	Поставщик расширенных сервисов
Create a new capability profile based on an existing capability profile template	Создание нового профиля возможностей путем использования существующего профиля возможностей
requestExistingTemplate (template ID)	***requestExistingTemplate*** (идентификатор шаблона)
returnExistingTemplate (existing template, processing error)	***returnExistingTemplate*** (существующий шаблон, ошибка обработки)
processFilledProfile (profile ID)	***processFilledProfile*** (идентификатор профиля)
returnProcessingResult(ID check error, storage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)
Create a new capability profile based on an existing capability profile	Создание нового профиля путем использования существующего профиля возможностей
requestExistingProfile (profile ID)	***requestExistingProfile*** (идентификатор профиля)
returnExistingProfile (existing profile, processing error)	***returnExistingProfile*** (существующий профиль, ошибка обработки)
processModifiedProfile (profile ID)	***processModifiedProfile*** (идентификатор профиля)
returnProcessingResult(ID check error, storage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)

Примечание 1 — Пунктирная линия разделяет два варианта создания профиля.

Примечание 2 — Созданию профиля всегда предшествует предварительный шаг регистрации и верификации уникального идентификатора шаблона, находящийся вне области применения ИСО 16100.

Рисунок 8 — Служба ***createProfile***

6.2.3 Служба *accessProfile*

6.2.3.1 Профиль ESI, отличный от профиля MSU

Служба ***accessProfile*** дает возможность пользователю профиля получить доступ к профилю возможностей MSU из профиля ESI, отличного от профиля MSU. Служба ***accessProfile*** использует, как минимум, службу ***requestExistingProfile*** и службу ***returnExistingProfile***. Служба ***accessProfile*** включает нижеследующие шаги:

a) пользователь профиля инициирует службу ***requestExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingProfile*** является идентификатор профиля;

b) поставщик сервисов инициирует службу ***returnExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingProfile*** являются существующий профиль и ошибка обработки.

На рисунке 9 (выше пунктирной линии) на языке UML приведена диаграмма последовательности обязательных шагов процедуры получения доступа к профилю, основанной на использовании идентификатора профиля ESI.

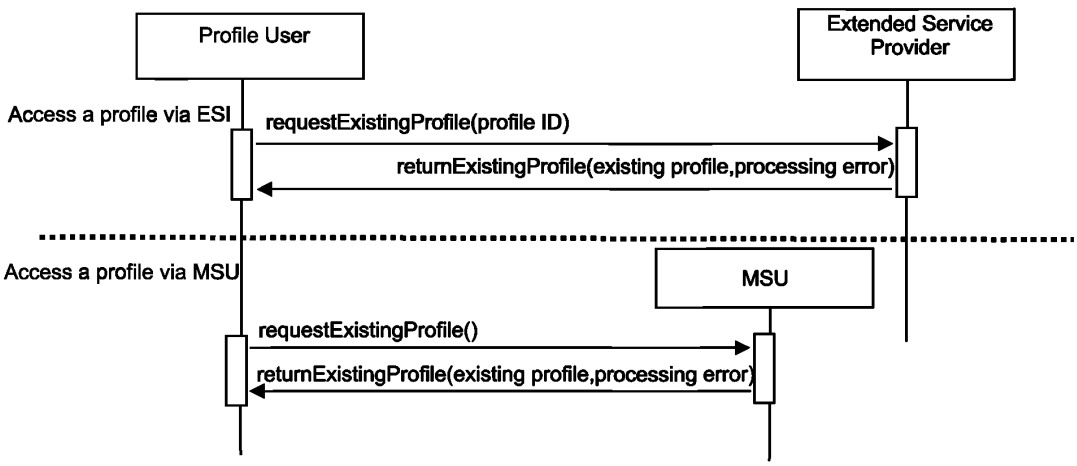
6.2.3.2 Профиль из MSU

Служба accessProfile дает возможность пользователю профиля получить доступ к требуемому профилю возможностей MSU. Служба ***accessProfile*** использует, как минимум, службу ***requestExistingProfile*** и службу ***returnExistingProfile***. Служба ***accessProfile*** включает нижеследующие шаги:

a) пользователь профиля инициирует службу ***requestExistingProfile*** объекта MSU, в котором параметры службы ***requestExistingProfile*** отсутствуют;

b) поставщик сервисов инициирует службу ***returnExistingProfile*** объекта MSU, в котором параметрами службы ***returnExistingProfile*** являются существующий профиль и ошибка обработки.

На рисунке 9 (ниже пунктирной линии) на языке UML приведена диаграмма последовательности обязательных шагов процедуры получения доступа к профилю, основанной на использовании идентификатора профиля MSU.



Template User	Пользователь профиля
Extended Service Provider	Поставщик расширенных сервисов
Access a profile via ESI	Получение доступа к профилю ESI
requestExistingProfile (profile ID)	***requestExistingProfile*** (идентификатор профиля)
returnExistingProfile (existing profile, processing error)	***returnExistingProfile*** (существующий профиль, ошибка обработки)
Access a profile via MSU	Получение доступа к профилю MSU
MSU	Блок программных средств организации производства
requestExistingProfile ()	***requestExistingProfile*** ()
returnExistingProfile (existing profile, processing error)	***returnExistingProfile*** (существующий профиль, ошибка обработки)

Рисунок 9 — Служба accessProfile

6.2.4 Служба *modifyProfile*

6.2.4.1 Профиль, доступный в интерфейсе ESI

Служба ***modifyProfile*** использует службу ***requestExistingProfile,*** службу ***returnExistingProfile,*** службу ***processModifiedProfile*** и службу ***returnProcessingResult***. Она дает возможность пользователю профиля модифицировать существующий профиль, доступный в интерфейсе ESI. Служба ***modifyProfile*** включает нижеследующие шаги:

a) пользователь профиля инициирует службу ***requestExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingProfile*** является идентификатор профиля;

b) поставщик сервисов инициирует службу ***returnExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingProfile*** являются существующий профиль и ошибка обработки;

c) пользователь профиля модифицирует существующий профиль, используя объект MDD модели MDM, а затем инициирует службу ***processModifiedProfile*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***processModifiedProfile*** является идентификатор профиля;

d) поставщик сервисов проверяет уникальность идентификатора профиля, а затем инициирует службу ***returnProcessingResult*** объекта ***ServiceAccessPoint***, в котором параметрами услуги ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 10 (выше пунктирной линии) на языке UML приведена диаграмма последовательности процедуры модификации профиля с помощью существующего профиля ESI.

6.2.4.2 Профиль, доступный в MSU

Служба modifyProfile использует службу ***requestExistingProfile,*** службу ***returnExistingProfile,*** службу ***processModifiedProfile*** и службу ***returnProcessingResult.*** Она дает возможность пользователю профиля модифицировать существующий профиль, доступный в MSU. Служба ***modifyProfile*** включает нижеследующие шаги:

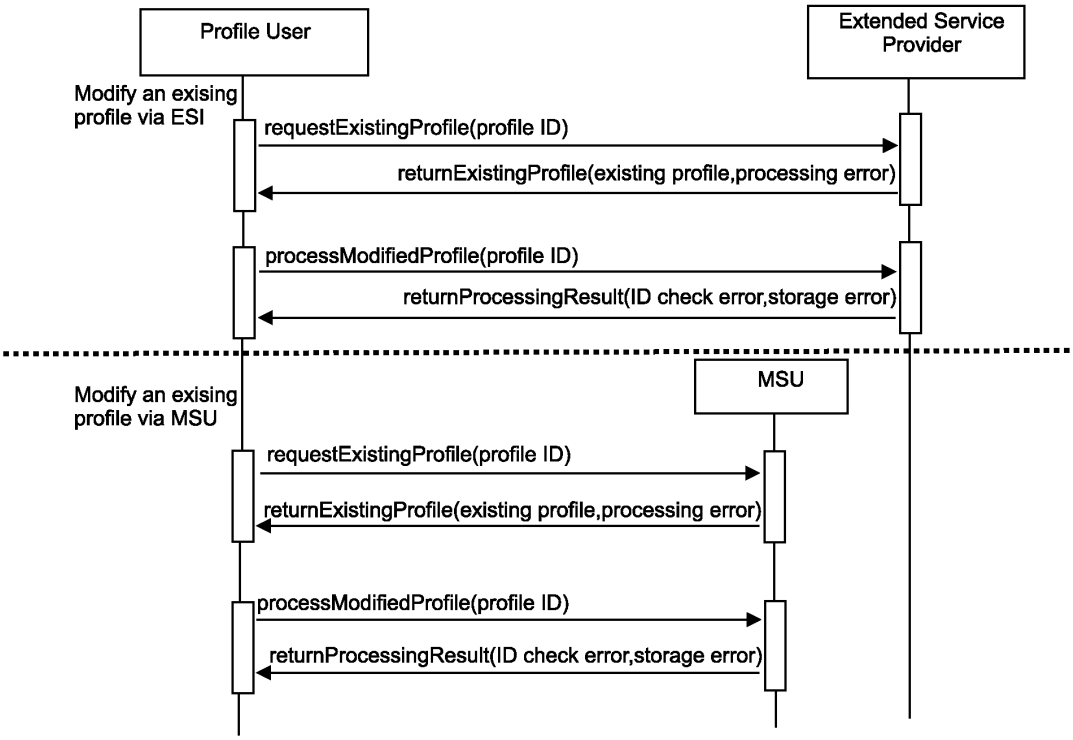
a) пользователь профиля инициирует службу ***requestExistingProfile*** объекта MSU, в котором параметром службы ***requestExistingProfile*** является идентификатор профиля;

b) поставщик сервисови инициирует службу ***returnExistingProfile*** объекта MSU, в котором параметрами службы ***returnExistingProfile*** являются существующий профиль и ошибка обработки;

c) пользователь профиля модифицирует существующий профиль, используя объект данных MDD модели MDM, а затем инициирует службу ***processModifiedProfile*** объекта MSU, в котором параметром службы ***processModifiedProfile*** является идентификатор профиля;

d) поставщик сервисов проверяет уникальность идентификатора профиля, а затем инициирует службу ***returnProcessingResult*** объекта MSU, в котором параметрами услуги ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 10 (ниже пунктирной линии) на языке UML приведена диаграмма последовательности процедуры модификации профиля с помощью существующего профиля через MSU.



Template User	Пользователь профиля
Extended Service Provider	Поставщик расширенных сервисов
Modify an existing profile via ESI	Модификация расширенного профиля через ESI
requestExistingProfile (profile ID)	***requestExistingProfile*** (идентификатор профиля)
returnExistingProfile (existing profile, processing error)	***returnExistingProfile*** (существующий профиль, ошибка обработки)
processModifiedProfile (profile ID)	***processModifiedProfile*** (идентификатор профиля)
returnProcessingResult (ID check error, storage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)
Modify an existing profile via MSU	Модификация существующего профиля через MSU
MSU	MSU=Блок программных средств организации производства
requestExistingProfile (profile ID)	***requestExistingProfile*** (идентификатор профиля)
returnExistingProfile (existing profile, processing error)	***returnExistingProfile*** (существующий профиль, ошибка обработки)
processModifiedProfile (profile ID)	***processModifiedProfile*** (идентификатор профиля)
returnProcessingResult (ID check error, storage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)

Рисунок 10 — Служба modifyProfile

6.2.5 Служба *validateProfile*

Служба ***validateProfile*** использует службу ***requestExistingProfile***, службу ***returnExistingProfile***, службу ***testProfile***, и службу ***returnTestResult.*** Она дает возможность пользователю профиля провести проверку соответствия существующего профиля.

Служба ***validateProfile*** включает нижеследующие шаги:

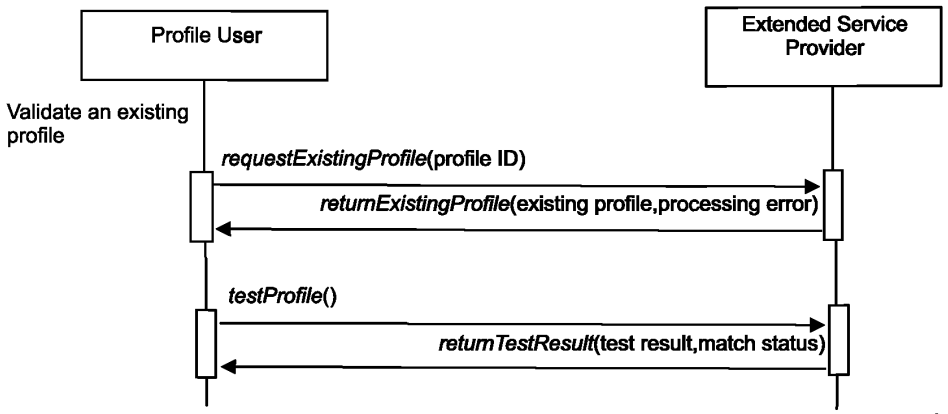
a) пользователь профиля инициирует службу ***requestExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingProfile*** является идентификатор профиля;

b) поставщик сервисов инициирует службу ***returnExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingProfile*** являются существующий профиль и ошибка обработки;

c) пользователь профиля инициирует службу ***testProfile*** объекта ***ServiceAccessPoint***, в котором параметры, ассоциированные со службой ***testProfile***, отсутствуют;

d) поставщик сервисов инициирует службу ***returnTestResult*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnTestResult*** являются результат испытаний и статус сопоставления.

На рисунке 11 на языке UML приведена диаграмма обязательных шагов последовательности процедуры выполнения проверки соответствия существующего профиля, основанной на использовании идентификатора профиля.



Template User	Пользователь профиля
Extended Service Provider	Поставщик расширенных сервисов
Validate an existing profile	Валидация существующего профиля
requestExistingProfile (profile ID)	***requestExistingProfile*** (идентификатор профиля)
returnExistingProfile (existing profile, processing error)	***returnExistingProfile*** (существующий профиль, ошибка обработки)
testProfile ()	***testProfile*** ()
returnTestResult (test result, match status)	***returnTestResult*** (результат испытаний, статус сопоставления)

Рисунок 11 — Служба validateProfile

6.2.6 Служба *deleteProfile*

Служба ***deleteProfile*** использует службу ***requestExistingProfile***, службу ***returnExistingProfile***, службу ***removeProfile*** и службу ***returnRemoveResult***. Она дает возможность пользователю профиля стереть существующий профиль в Архиве посредством ESI.

Служба ***deleteProfile*** включает нижеследующие шаги:

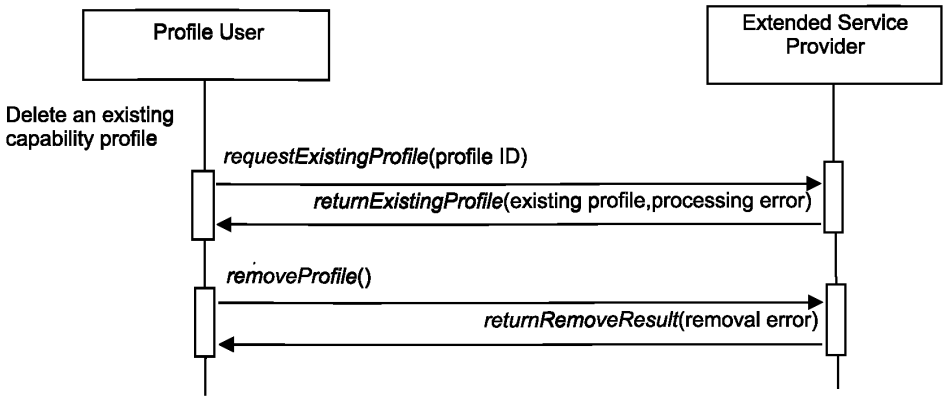
a) пользователь профиля инициирует службу ***requestExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingProfile*** является идентификатор профиля;

b) поставщик сервисов инициирует службу ***returnExistingProfile*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingProfile*** являются существующий профиль и ошибка обработки;

c) пользователь профиля инициирует службу ***removeProfile*** объекта ***ServiceAccessPoint***, в котором параметры, ассоциированные с службой ***removeProfile***, отсутствуют;

d) поставщик сервисов инициирует службу returnRemoveResult объекта ***ServiceAccessPoint***, в котором параметром службы returnRemoveResult является ошибка удаления.

На рисунке 12 на языке UML приведена диаграмма последовательности обязательных шагов процедуры стирания профиля, основанной на использовании идентификатора профиля.



Template User	Пользователь профиля
Extended Service Provider	Поставщик расширенных сервисов
Delete an existing capability profile	Стирание существующего профиля возможностей
requestExistingProfile (profile ID)	***requestExistingProfile*** (идентификатор профиля)
returnExistingProfile (existing profile, processing error)	***returnExistingProfile*** (существующий профиль, ошибка обработки)
returnRemoveResult (removal error)	***returnRemoveResult*** (ошибка удаления)

Рисунок 12 — Служба ***deleteProfile***

6.3 Группа CCSI

6.3.1 Сценарии, используемые группой CCSI

Группа CCSI использует нижеследующие сценарии шаблона профиля возможностей:

a) создание CCS (структуры класса возможностей) либо путем заполения заготовки шаблона CCS, либо путем модификации существующего CCS, а затем получение созданного CCS от поставщика сервисов;

a) получение доступа к CCS в Архиве с помощью идентификатора CCS, а затем получение CCS от поставщика сервисов;

b) модификация CCS в соответствии либо с требованиями пользователя, либо с результатами проверки соответствия, а затем получение модифицированного CCS от поставщика сервисов;

c) испытание CCS на соответствие установленным критериям соответствия CCS и получение либо положительного отклика, либо отрицательного отклика, либо отклика уровня сопоставления от поставщика сервисов;

d) регистрация CCS в Архиве и получение статуса «зарегистрирован» от поставщика сервисов;

e) удаление CCS с помощью идентификатора идентификатор CCS и получение статуса “удален” от поставщика сервисов.

6.3.2 Создание структуры класса возможностей

6.3.2.1 Процесс создания

Процесс создания структуры класса возможностей начинается с анализа рассматриваемого производственного приложения. Производственный процесс, состоящий из набора операций, - ключевой объект настоящего анализа. Указанные операции представлены узлами структуры класса возможностей. Путем создания новых узлов, пользователь может создавать новые структуры класса возможностей внутри той же модели MDM.

Процесс создания структуры класса возможностей включает либо заполнение заготовки шаблона CCS, либо модификацию существующей CCS в соответствии с рассматриваемым приложением.

6.3.2.2 Служба ***createCCS***

6.3.2.2.1 Создание CCS из заготовки шаблона CCS

Служба ***createCCS*** дает возможность пользователю CCS создавать новые CCS путем заполнения заготовки шаблона CCS. При создании CCS из заготовки шаблона CCS, служба ***createCCS*** использует, как минимум, службу ***requestBlankCCS***, службу ***returnBlankCCS***, службу ***processFilledCCS*** и службу ***returnProcessingResult.***

Служба createCCS, предназначенная для создания новой структуры CCS из заготовки шаблона CCS, включает нижеследующие шаги:

a) пользователь CCS инициирует службу ***requestBlankCCS*** объекта ***ServiceAccessPoint***, в которой параметры, ассоциированные со службой ***requestBlankCCS,*** отсутствуют;

b) поставщик сервисов инициирует службу ***returnBlankCCS*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnBlankCCS*** являются заготовка CCS и ошибка создания;

c) пользователь CCS заполняет заготовку шаблона CCS, используя объекты MDD модели MDM, а затем инициирует службу ***processFilledCCS*** объекта ***ServiceAccessPoint***, в котором параметром службы ***processFilledCCS*** является идентификатор CCS;

d) поставщик сервисов проверяет уникальность идентификатора CCS, а затем инициирует службу ***returnProcessingResult*** объекта ***ServiceAccessPoint,*** в котором параметрами услуги ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 13 (выше пунктирной линии) на языке UML приведена диаграмма последовательности обязательных шагов процедуры создания новой CCS из формальной структуры CCS.

6.3.2.2.2 Структура CCS, созданная путем модификации существующей структуры CCS

Служба ***createCCS*** дает возможность пользователю CCS создавать новые структуры CCS путем модификации существующих CCS. При создании новых CCS путем модификации существующих CCS, служба ***createCCS*** использует, как минимум, службу ***requestExistingCCS,*** службу ***returnExistingCCS,*** службу ***processModifiedCCS*** и службу ***returnProcessingResult***. Служба ***createCCS*** включает нижеследующие шаги:

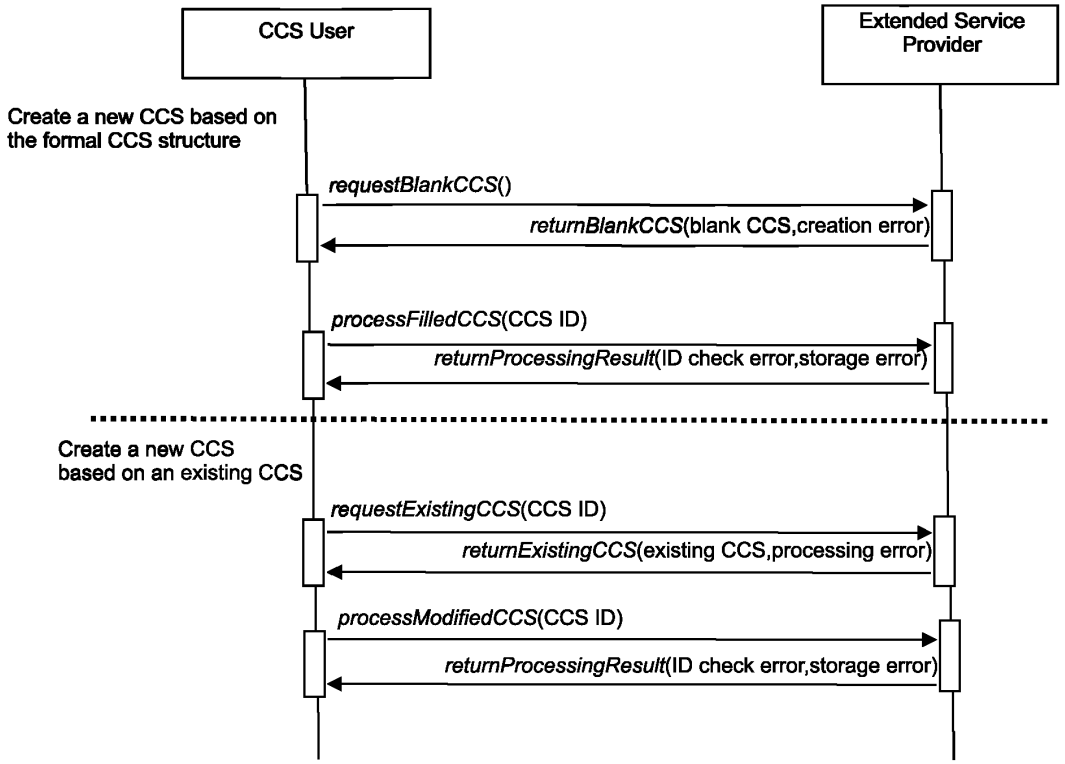
a) пользователь CCS инициирует службу ***requestExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingCCS*** является идентификатор CCS;

b) поставщик сервисов инициирует службу ***returnExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingCCS*** являются существующая CCS и ошибка обработки;

c) пользователь CCS модифицирует существующую CCS а затем инициирует службу ***processModifiedCCS*** объекта ***ServiceAccessPoint***, в котором параметром службы ***processModifiedCCS*** является идентификатор CCS;

d) поставщик сервисов проверяет уникальность идентификатора CCS, а затем инициирует службу ***returnProcessingResult*** объекта ***ServiceAccessPoint***, в котором параметрами услуги ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 13 (ниже пунктирной линии) на языке UML приведена диаграмма последовательности обязательных шагов процедуры создания новой структуры CCS из существующей структуры CCS.



CCS User	Пользователь структуры CCS
Extended Service Provider	Поставщик расширенных сервисов
Create a nem CCS based on the formal CCS structure	Создание новой CCS на основе формальной структуры CCS
requestBlankCCS()	***requestBlankCCS*** ()
returnBlankCCS(blank CCS, creation error)	***returnBlankCCS*** (заготовка CCS, ошибка создания)
processFilliedCCS (CCS ID)	***processFilliedCCS*** (идентификатор CCS)
returnProcessingResult(ID check error, storage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)
Create a new CCS based on an existing CCS	Создание новой CCS на основе существующей CCS
requestExistingCCS(CCS ID)	***requestExistingCCS*** (идентификатор CCS)
returnExistingCCS(existing CCS, processing error)	***returnExistingCCS*** (существующая CCS, ошибка обработки)
processModifiedCCS(CCS ID)	***processModifiedCCS*** (идентификатор CCS)

Примечание 1 — Пунктирная линия разделяет два варианта создания сущности.

Примечание 2 — Созданию CCS всегда предшествует предварительный шаг регистрации и верификации уникального идентификатора CCS, находящийся вне области применения ИСО 16100.

Рисунок 13 — Служба ***createCCS***

6.3.3 Служба *accessCCS*

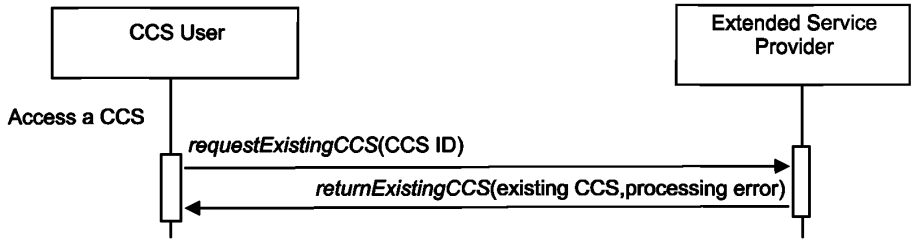
Служба ***accessCCS*** использует службу ***requestExistingCCS*** и службу ***returnExistingCCS***. Она дает возможность пользователю CCS получить доступ к существующей CCS.

Служба ***accessCCS*** включает нижеследующие шаги:

a) пользователь CCS инициирует службу ***requestExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingCCS*** является идентификатор CCS;

b) поставщик сервисов инициирует службу ***returnExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingCCS*** являются существующая CCS и ошибка обработки.

На рисунке 14 на языке UML приведена диаграмма последовательности обязательных шагов процедуры получения доступа к CCS, использующей идентификатор CCS.



CCS User	Пользователь структуры CCS
Extended Service Provider	Поставщик расширенных сервисов
Access a CCS	Получение доступа к CCS
requestExistingCCS(CCS ID)	***requestExistingCCS*** (идентификатор CCS)
returnExistingCCS(existing CCS, processing error)	***returnExistingCCS*** (существующая CCS, ошибка обработки)

Рисунок 14 — Служба ***accessCCS***

6.3.4 Служба *modifyCCS*

Служба ***modifyCCS*** использует службу ***requestExistingCCS***, службу ***returnExistingCCS***, службу ***processModifiedCCS*** и службу ***returnProcessingResult***. Она дает возможность пользователю CCS модифицировать существующую CCS.

Служба ***modifyCCS*** включает нижеследующие шаги:

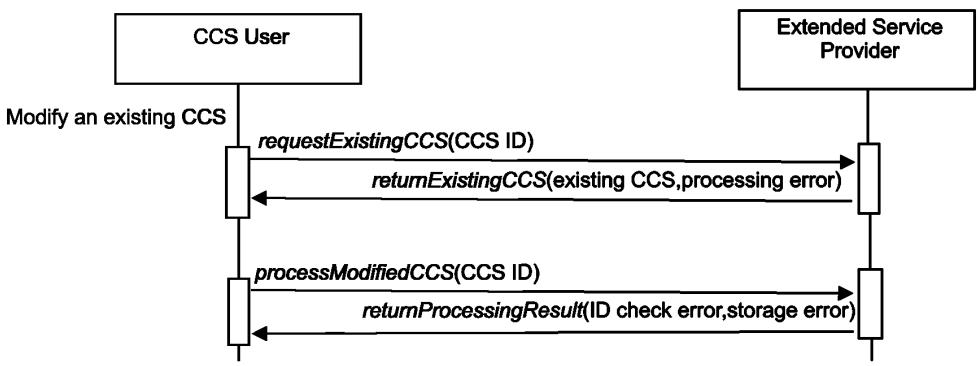
a) пользователь CCS инициирует службу ***requestExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingCCS*** является идентификатор CCS;

b) поставщик сервисов инициирует службу ***returnExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingCCS*** являются существующая CCS и ошибка обработки;

c) пользователь CCS модифицирует существующую CCS, а затем инициирует службу ***processModifiedCCS*** объекта ***ServiceAccessPoint***, в котором параметром службы ***processModifiedCCS*** является идентификатор CCS;

d) поставщик сервисов проверяет уникальность идентификатора CCS, а затем инициирует службу ***returnProcessingResult*** объекта ***ServiceAccessPoint***, в котором параметрами услуги ***returnProcessingResult*** являются ошибка проверки идентификатора и ошибка хранения.

На рисунке 15 на языке UML приведена диаграмма последовательности обязательных шагов процедуры модификации существующей CCS на основе идентификатора CCS.



CCS User	Пользователь структуры CCS
Extended Service Provider	Поставщик расширенных сервисов
Modify an existing CCS	Модификация существующей CCS
requestExistingCCS(CCS ID)	***requestExistingCCS*** (идентификатор CCS)
returnExistingCCS(existing CCS, processing error)	***returnExistingCCS*** (существующая CCS, ошибка обработки)
processModifiedCCS(CCS ID)	***processModifiedCCS*** (идентификатор CCS)
returnProcessingResult(ID check error, storage error)	***returnProcessingResult*** (ошибка проверки идентификатора, ошибка хранения)

Рисунок 15 — Служба modifyCCS

6.3.5 Служба *validateCCS*

Служба ***validateCCS*** использует службу ***requestExistingCCS***, службу ***returnExistingCCS***, службу ***testCCS*** и службу ***returnTestResult***. Она дает возможность пользователю CCS провести проверку соответствия существующей CCS.

Служба ***validateCCS*** включает нижеследующие шаги:

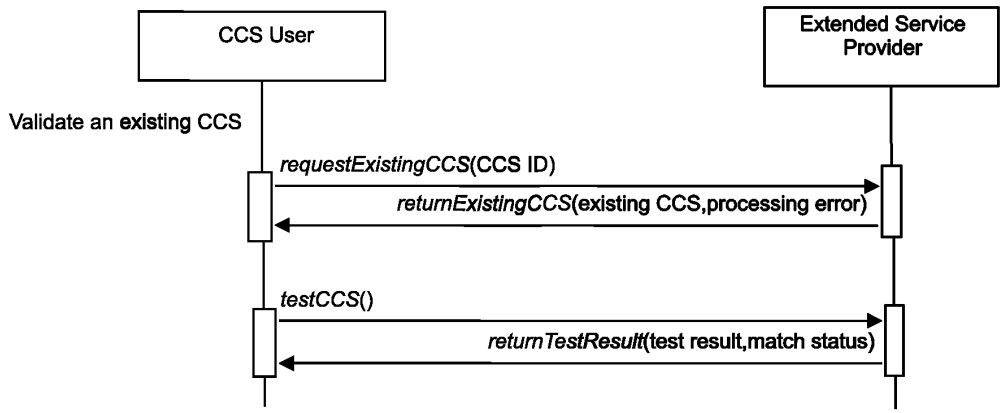
a) пользователь CCS инициирует службу ***requestExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingCCS*** является идентификатор CCS;

b) поставщик сервисов инициирует службу ***returnExistingCCS*** объекта ***ServiceAccessPoint,*** в котором параметрами службы ***returnExistingCCS*** являются существующая CCS и ошибка обработки;

c) пользователь CCS инициирует службу ***testCCS*** объекта ***ServiceAccessPoint***, в котором параметры, ассоциированные с услугой ***testCCS,*** отсутствуют;

d) поставщик сервисов инициирует службу ***returnTestResult*** объекта ***ServiceAccessPoint,*** в котором параметрами услуги ***returnTestResult*** являются результат испытаний и ошибка статуса.

На рисунке 16 на языке UML приведена диаграмма последовательности обязательных шагов процедуры проверки соответствия существующей CCS с помощью идентификатора CCS.



CCS User	Пользователь CCS
Extended Service Provider	Поставщик расширенных сервисов
Validate an existing CCS	Валидация существующей CCS
requestExistingCCS(CCS ID)	***requestExistingCCS*** (идентификатор CCS)
returnExistingCCS(existing CCS, processing error)	***returnExistingCCS*** (существующая CCS, ошибка обработки)
testCCS( )	testCCS ()
returnTestResult(test result, error status)	***returnTestResult*** (результат испытаний, ошибка статуса)

Рисунок 16 — Служба validateCCS

6.6.6 Служба *deleteCCS*

Служба ***deleteCCS*** использует службу ***requestExistingCCS***, службу ***returnExistingCCS***, службу ***removeCCS*** и службу ***returnRemoveResult***. Она дает возможность пользователю CCS стереть существующую CCS.

Служба ***deleteCCS*** включает нижеследующие шаги:

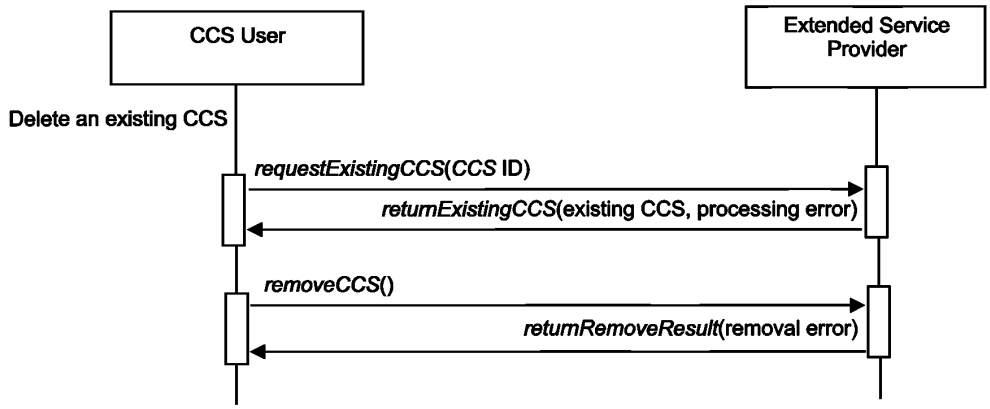
a) пользователь CCS инициирует службу ***requestExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметром службы ***requestExistingCCS*** является идентификатор CCS;

b) поставщик сервисов инициирует службу ***returnExistingCCS*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnExistingCCS*** являются существующая CCS и ошибка обработки;

c) пользователь CCS инициирует службу ***removeCCS*** объекта ***ServiceAccessPoint***, в котором параметры, ассоциированные со службой ***removeCCS***, отсутствуют;

d) поставщик сервисов инициирует службу ***returnRemoveResult*** объекта ***ServiceAccessPoint***, в котором параметром службы ***returnRemoveResult*** является ошибка удаления.

На рисунке 17 на языке UML приведена диаграмма последовательности обязательных шагов процедуры стирания CCS с помощью идентификатора CCS.



CCS User	Пользователь CCS
Extended Service Provider	Поставщик расширенных сервисов
Delete an existing CCS	Стирание существующей CCS
requestExistingCCS(CCS ID)	***requestExistingCCS*** (идентификатор CCS)
returnExistingCCS(existing CCS, processing error)	***returnExistingCCS*** (существующая CCS, ошибка обработки)
removeCCS ()	***removeCCS*** ()
returnRemoveResult(removal error)	***returnRemoveResult*** (ошибка удаления)

Рисунок 17 — Служба deleteCCS

6.4 Расширенная группа обнаружения совпадений

6.4.1 Сценарии, используемые расширенной группой обнаружителей совпадений

Расширенная группа обнаружения совпадений использует нижеследующие сценарии получения доступа к двум профилям возможностей и их сопоставления:

a) запрос профиля возможностей MSU либо из Архива, в котором данный профиль возможностей MSU зарегистрирован, либо из MSU, с которым данный профиль возможностей MSU ассоциирован, а затем получение профиля возможностей MSU;

b) сопоставление двух профилей возможностей (существующего профиля возможностей MSU и требуемого профиля возможностей) и получение результата сопоставления, включающего уровень сопоставления(см. ИСО 16100-5:2009, раздел 7.2) и отчет о согласованных и несогласованных функциях от поставщика сервисов.

6.4.2 Служба *ExtendedMatcher*

Служба ***ExtendedMatcher*** использует службу ***requestProfile***, службу ***returnProfile***, службу ***requestMatching*** и службу ***returnMatchingResult***. Она дает возможность пользователю запросить профиль либо из архива, либо из MSU (блока программных средств организации производства), и сопоставить профиль возможностей MSU с требуемым профилем возможностей, используя рассматриваемый обнаружитель совпадений.

Служба ***ExtendedMatcher*** включает нижеследующие шаги:

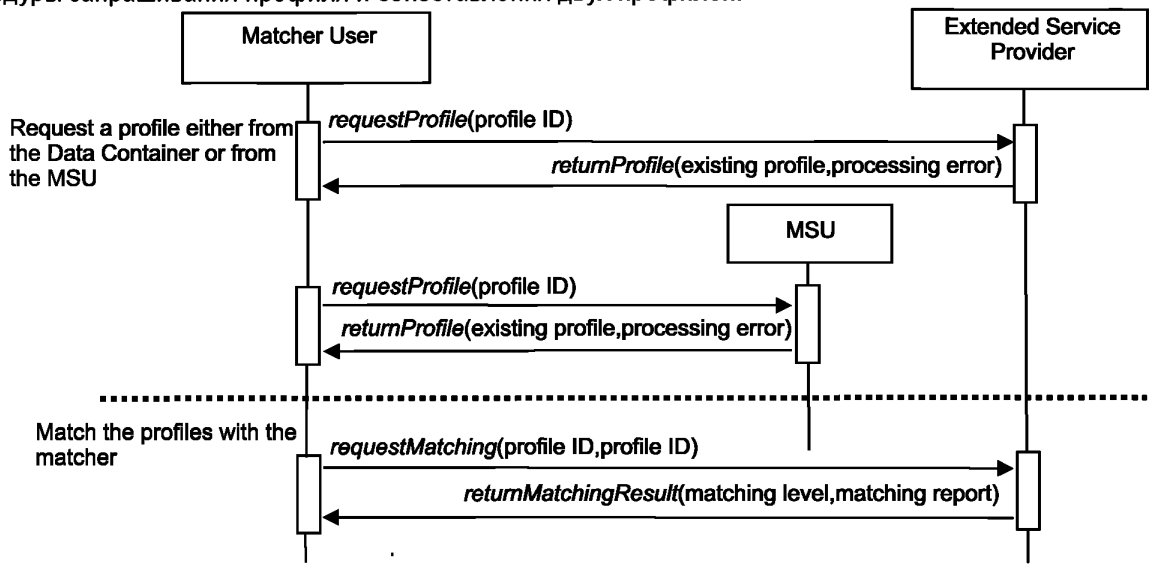
a) пользователь обнаружителя совпадений инициирует службу ***requestProfile*** объекта ***ServiceAccessPoint*** или объекта MSU, в котором параметром службы ***requestProfile*** является идентификатор профиля;

b) поставщик сервисов инициирует службу ***returnProfile*** объекта ***ServiceAccessPoint*** или объекта MSU, в котором параметрами службы ***returnProfile*** являются существующий профиль и ошибка обработки;

c) пользователь обнаружителя совпадений инициирует службу ***requestMatching*** объекта ***ServiceAccessPoint***, в котором имеются два параметра идентификатора профиля;

d) поставщик сервисов инициирует службу ***returnMatchingResult*** объекта ***ServiceAccessPoint***, в котором параметрами службы ***returnMatchingResult*** являются уровень сопоставления (см. ИСО 16100-5:2009, раздел 7.2) и отчет о сопоставлении (отчет о согласованных и несогласованных функциях).

На рисунке 18 на языке UML приведена диаграмма последовательности обязательных шагов процедуры запрашивания профиля и сопоставления двух профилей.



Matcher User	Пользователь обнаружителя совпадений
Extended Service Provider	Поставщик расширенных сервисов
Request a profile either from the Data Container or from the MSU	Запрос профиля либо из контейнера данных, либо из блока программных средств организации производства
requestProfile(profile ID)	***requestProfile*** (идентификатор профиля)
returnProfile(existing profile, processing error)	***returnProfile*** (существующий профиль, ошибка обработки)
MSU	Блок программных средств организации производства
requestProfile(profile ID)	***requestProfile*** (идентификатор профиля)
returnProfile(existing profile, processing error)	***returnProfile*** (существующий профиль, ошибка обработки)
Match the profile with the matcher	Сопоставление профилей с помощью обнаружителя совпадений
requestMatching(profile ID, profile ID)	***requestMatching*** (идентификатор профиля, идентификатор профиля)
returnMatchingResult (matching level, matching report)	***returnMatchingResult*** (уровень сопоставления, отчет о сопоставлении)

Примечание — Пунктирная линия отделяет запрос профиля от запроса сопоставления.

Рисунок 18 — Служба ***ExtendedMatcher***
Рисунок 18, лист 2.

6.4.3 Проверка соответствия обнаружителя совпадений

Методология установления соответствия, описанная в ИСО 16100-4, используется в настоящем стандарте. Здесь утверждения соответствия для практической реализации CSI расширяются и используются обнаружителями совпадений профилей возможностей в соответствии со ИСО 16100-4:2006, таблица 9. Таблицы 1 и 2 в настоящем стандарте используются процедурой проверки соответствия. Типы точек соответствия, указанные в таблицах 1 и 2, определены в ИСО 16100-4:2006, таблица 5.

Таблица 1 — Утверждения соответствия CSI, необходимые для функционирования обнаружителей совпадений

Точка соответствия и № набора	Описание точки соответствия	Ссылка на спецификацию	Тип точки соответствия[a]	Абстрактный критерий испытания
Index_1	Получение требуемого профиля возможностей	ИСО 16100-5:2009, раздел 7	A	Верификация требуемого профиля возможностей на соответствие с ИСО 16100-5:2009, таблица 2
Index_2	Получение профиля возможностей MSU	ИСО 16100-5:2009, раздел 7	A	Верификация профиля возможностей MSU на соответствие с ИСО 16100-5:2009, таблица 2

Продолжение таблицы 1

Точка соответствия и № набора	Описание точки соответствия	Ссылка на спецификацию	Тип точки соответствия[a]	Абстрактный критерий испытания
Index_3	Сравнение идентификаторов двух профилей модели MDM	ИСО 16100-5:2009, раздел 7	**A**	—
Index_3.1	Извлечение идентификаторов из двух профилей модели MDM	ИСО 16100-5:2009, раздел 7	**A**	Верификация идентификаторов словарей
Index_3.2	Сравнение двух идентификаторов модели MDM	ИСО 16100-5:2009, раздел 7	**A**	Верификация и показ результатов сравнения
Index_3.2.1	Отчет о результате № 1 и переход к обнаружителю совпадений типа 1	ИСО 16100-5:2009, раздел 7	**A**	Верификация: 1) «Идентификаторы совпадают» 2) «Переход к обнаружителю совпадений типа 1»
Index_3.2.2	Отчет о результате № 2	ИСО 16100-5:2009, раздел 7	**A**	Верификация: «Идентификаторы не совпадают»
Index_4	Сравнение названий двух профилей модели MDM	ИСО 16100-5:2009, раздел 7	**A**	—
Index_4.1	Извлечение названий двух профилей модели MDM	ИСО 16100-5:2009, раздел 7	**A**	Верификация названий модели MDM
Index_4.2	Сравнение двух названий модели MDM	ИСО 16100-5:2009, раздел 7	**A**	Верификация результатов на основе сравнения
Index_4.2.1	Отчет о результате № 1 и завершение сопоставлени	ИСО 16100-5:2009, раздел 7	**A**	Верификация: 1) «Названия модели MDM не совпадают» 2) «Два профиля относятся к различным моделям MDM» 3) «Нет сопоставления»
Index_4.2.2	Отчет о результате № 2	ИСО 16100-5:2009, раздел 7	**A**	Верификация: «Названия модели MDM совпадают»
Index_5	Сравнение форматов определения возможностей двух профилей	ИСО 16100-5:2009, раздел 7	**A**	—
Index_5.1	Извлечение форматов определения возможностей из двух профилей	ИСО 16100-5:2009, раздел 7	**A**	Верификация форматов определения возможностей
Index_5.2	Сравнение форматов определения возможностей	ИСО 16100-5:2009, раздел 7	**A**	Верификация результатов на основе сравнения
Index_5.2.1	Отчет о результате № 1	ИСО 16100-5:2009, раздел 7	**A**	Показать отчет № 1: 1) «Форматы определения возможностей не совпадают» 2) «Преобразовать объекты MDD и MDD (определения возможностей) к единому формату определения возможностей»
Index_5.2.2	Отчет о результате № 2	ИСО 16100-5:2009, раздел 7	**A**	Верификация: «Форматы определения возможностей совпадают»
Index_6	Сравнение двух определений возможностей	ИСО 16100-5:2009, раздел 7	**A**	—
Index_6.1	Сравнение содержания элемента набора MDD элементов «**Set_of_MDD_object**» в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	—

Окончание таблицы 1

Точка соответствия и № набора	Описание точки соответствия	Ссылка на спецификацию	Тип точки соответствия[a]	Абстрактный критерий испытания
Index_6.1.1	Сравнение атрибутов «название» и «операция» элемента имени MDD **«MDD_name»** в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	Верификация результата
Index_6.2	Сравнение содержания двух списков объектов **«List_of_MDD_object»** в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	—
Index_6.2.1	Сравнение последовательностей элементов **«MDD_name»** в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	Верификация результата
Index_6.2.2	Сравнение атрибутов «название» и «операция» элемента **«MDD_name»**, находящихся в одинаковых позициях в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	Верификация результата
Index_6.3	Сравнение содержания двух упорядоченных во времени объектов **«Time_Ordered_MDD_objects»** в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	—
Index_6.3.1	Сравнение временной упорядоченности элементов **«MDD_name»** в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	Верификация результата
Index_6.3.2	Сравнение атрибутов «название» и «операция» в двух элементах **«MDD_name»**, находящихся в одинаковых позициях в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	Верификация результата
Index_6.4	Сравнение объектов **«Event_Ordered_MDD_object»**, расположенных в порядке поступления	ИСО 16100-5:2009, раздел 7	**A**	—
Index_6.4.1	Сравнение элементов **«MDD_name»**, расположенных в порядке поступления в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	Верификация результата
Index_6.4.2	Сравнение атрибутов «название» и «операция» в двух элементах **«MDD_name»**, находящихся в одинаковых позициях в двух определениях возможностей	ИСО 16100-5:2009, раздел 7	**A**	Верификация результата

[a] См. ИСО 16100-4:2006, таблица 5.

Таблица 2 — Использование утверждений соответствия CSI в отчетах обнаружителей совпадений

Точка соответствия и № набора	Описание точки соответствия	Ссылка на спецификацию	Тип точки соответствия[a]	Абстрактный критерий испытания
Index_1	MatchingLevelReport	ИСО 16100-5:2009, раздел 7.2	**A**	Допустимый уровень сопоставления: Полное совпадение Совпадают все обязательные элементы Совпадают некоторые обязательные элементы Обязательные элементы не совпадают
Index_2	DetaitedListReport	ИСО 16100-5:2009, раздел 7.2	**A**	Верификация согласованных функций и несогласованных функций
Index_3	CompareMatchingLevel & Detailed Report MatchingLevel	—	—	—
Index_3.1	ForCompleteMatch	—	—	Верификация того, что оба набора функций в двух профилях полностью эквивалентны
Index_3.2	ForAllMandatoryMatch	—	—	Верификация того, что оба набора обязательных функций в двух профилях полностью эквивалентны
Index_3.3	ForSomeMandatoryMatch	—	—	Верификация списка эквивалентных обязательных функций в двух профилях и списка неэквивалентных обязательных функций
Index_3.4	ForNoMandatoryMatch	—	—	Настоящие оба набора функций в двух профилях рассматриваются как полностью неэквивалентные

[a] См. ИСО 16100-4:2006, таблица 5.

7 Формальное описание протокола расширенных сервисов ESI

7.1 Синтаксис основных сервисов

Синтаксис услуги унифицированного названия ресурса URN, описанный в ИСО 16100-3, используется в настоящем стандарте.

Характеристическое название услуги типа URN начинается с строки «услуга:». Настоящее название услуги типа URN включает тип сервиса и соответствующую точку доступа услуги, в которую не включается разделитель «:», с которого начинается адрес спецификации. Информация об атрибуте рассматриваемой услуги следует за адресом спецификации, закодированным в соответствии с грамматикой универсального имени ресурса URN.

Полное имя URN услуги имеет нижеследующий синтаксис:

«service:<service-type>:<service-access-point>://<address>;<atribute-list>»

Элемент <**service-type**> в строке URN представляет характеристическую услугу, идентифицированную в разделе 5.1.

Элемент <**service-access-type**> в строке URN представляет собой точку доступа сервисные группы ESI. Данные сервисные группы идентифицированы в разделе 5.2.

Элемент <**address**> в строке URN задает путь к поставщику услуг ESI.

Список атрибутов включает знаки «;», разделяющие назначения атрибутов. Назначения атрибутов имеют форму:

<**atribute-id**>=<**atribute-value**>

В дополнение, для ключевых атрибутов используется форма <**atribute-id**>.

Подробное описание услуг на языке UML во всех диаграммах в Разделе 6 использует формальный синтаксис в соответствии с оставшимися подразделами раздела 7.

7.2 Служебный протокол CPTI группы

7.2.1 Создание шаблона профиля возможностей

7.2.1.1 Создание с помощью формальной структуры

Служба ***createTemplate*** генерирует шаблон с помощью формальной структуры. Она включает нижеследующие шаги:

a) служба ***requestBlankTemplate*** запрашивает заготовку шаблона и задает тип сервиса:

<service-type>=*"requestBlankTemplate"*

b) служба ***returnBlankTemplate*** получает заготовку шаблона и задает тип сервиса:

<service-type>=*"returnBlankTemplate"* с соответствующими атрибутами:

- template_content="the_template_content"

- access_status="the_access_status"

c) служба ***processFilledTemplate*** запрашивает приемлемость заполненного шаблона и задает тип сервиса:

<*service-type*>="*processFilledTemplate*" с соответствующими атрибутами:

- template_ID="the_template_id"

d) служба ***returnProcessingResult*** получает результат обработки и задает тип сервиса:

<service-type>="*returnProcessingResult*" с соответствующими атрибутами:

- ID_check_error="ID_check_error"

- storage_error="storage_error"

7.2.1.2 Создание с помощью существующего шаблона

Служба ***createTemplate*** также генерирует шаблон с помощью существующего шаблона. Она включает нижеследующие шаги:

a) Служба ***requestExistingTemplate*** запрашивает существующий шаблон и задает тип сервиса:

<service-type>="*requestExistingTemplate*" с соответствующими атрибутами:

- template_ID="the_template_id";

b) Служба ***returnExistingTemplate*** получает существующий шаблон и задает тип сервиса:

<service-type>="*returnExistingTemplate*" с соответствующими атрибутами:

- template_content="the_template_content"

- process_status="the_process_status"

c) Служба ***processModifiedTemplate*** запрашивает приемлемость модифицированного шаблона и задает тип сервиса:

<service-type>="*processModifiedTemplate*" с соответствующими атрибутами:

- template_ID="the_template_id"

d) Служба ***returnProcessingResult*** получает результат обработки и задает тип сервиса:

<service-type>="*returnProcessingResult*" с соответствующими атрибутами:

- ID_check_error="ID_check_error"

- storage_error="storage_error"

7.2.2 Доступ к шаблону профиля возможностей

Служба ***accessTemplate*** получает доступ к шаблону и включает нижеследующие шаги:

a) служба ***requestExistingTemplate*** получает доступ к существующему шаблону и задает тип сервиса:

<service-type>="*requestExistingTemplate*" с соответствующими атрибутами:

- template_ID="the_template_id";

b) служба ***returnExistingTemplate*** получает запрошенный шаблон. Тип сервиса:

<service-type>="*returnExistingTemplate*" с соответствующими атрибутами:

- template_content="the_template_content"

- process_status="the_process_status"

7.2.3 Модификация шаблона профиля возможностей

Служба ***modifyTemplate*** модифицирует шаблон и включает нижеследующие шаги:

a) служба ***requestExistingTemplate*** получает доступ к существующему шаблону и задает тип сервиса:

<service-type>="*requestExistingTemplate*" с соответствующими атрибутами:

- template_ID="the_template_id"

b) служба ***returnExistingTemplate*** получает запрошенный шаблон и задает тип сервиса:
<service-type>="*returnExistingTemplate*" с соответствующими атрибутами:
- template_content="the_template_content"
- process_status="the_process_status"
c) служба ***processModifiedTemplate*** запрашивает приемлемость модифицированного шаблона и задает тип сервиса:
<service-type>="*processModifiedTemplate*" с соответствующими атрибутами:
- template_ID="the_template_id"
d) служба ***returnProcessingResult*** получает результат обработки. Тип сервиса:
<service-type>="*returnProcessingResult*" с соответствующими атрибутами:
- ID_check_error="ID_check_error"
- storage_error="storage_error"

7.2.4 Проверка соответствия шаблона профиля возможностей

Служба ***validateTemplate*** обеспечивает испытание шаблона и включает нижеследующие шаги:
a) служба ***requestUnregisteredTemplate*** получает доступ к незарегистрированному шаблону и задает тип сервиса:
<service-type>="*requestUnregisteredTemplate*" с соответствующими атрибутами:
- template_ID="the_template_id"
b) служба ***returnUnregisteredTemplate*** получает запрошенный шаблон. Тип сервиса:
<service-type>="*returnUnregisteredTemplate*" с соответствующими атрибутами:
- template_content="the_template_content"
- process_status="the_process_status"
c) служба ***testTemplate*** верифицирует незарегистрированный шаблон. Тип сервиса:
<service-type>="*testTemplate*"
d) служба ***returnTestResult*** получает результат испытаний и статус испытаний. Тип сервиса:
<service-type>="*returnTestResult*" с соответствующими атрибутами:
- test_result ="the_test_result"
- test_status="the_test_status"

7.2.5 Стирание шаблона профиля возможностей

Служба deleteTemplate стирает существующий шаблон и включает нижеследующие шаги:
a) служба ***requestExistingTemplate*** получает доступ к существующему шаблону. Тип сервиса:
<service-type>="*requestExistingTemplate*" с соответствующими атрибутами:
- template_ID="the_template_id"
b) служба ***returnExistingTemplate*** получает запрошенный шаблон. Тип сервиса:
<service-type>="*returnExistingTemplate*" с соответствующими атрибутами:
- template_content= "the_template_content"
- process_status="the_process_status"
c) служба ***removeTemplate*** стирает шаблон профиля возможностей из архива. Тип сервиса:
<service-type>="*removeTemplate*"
d) служба ***returnRemoveResult*** получает статус удаления. Тип сервиса:
<service-type>="*returnRemoveResult*" с соответствующими атрибутами:
- remove_status="the_remove_status"

7.3 Служебные протоколы расширенной CPI группы

7.3.1 Создание профиля возможностей

7.3.1.1 Создание с помощью шаблона профиля возможностей
Служба ***createProfile*** генерирует новый профиль с помощью шаблона профиля возможностей и включает нижеследующие шаги:
a) служба ***requestExistingTemplate*** запрашивает существующий шаблон. Тип сервиса:
<service-type>="*requestExistingTemplate*" с соответствующими атрибутами:
- template_ID="the_template_id"
b) служба ***returnExistingTemplate*** получает запрошенный шаблон. Тип сервиса:
<service-type>="*returnExistingTemplate*" с соответствующими атрибутами:

- template_content="the_template_content"
- access_status="the_access_status"
c) служба ***processFilledProfile*** запрашивает приемлемость заполненного профиля. Тип сервиса:
<service-type>=”*processFilledProfile*” с соответствующими атрибутами:
- profile_ID=”the_profile_id”
d) служба ***returnProcessingResult*** получает результат обработки. Тип сервиса:
<service-type>="*returnProcessingResult*" с соответствующими атрибутами:
- ID_check_error="ID_check_error"
- storage_error="storage_error"
7.3.1.2 Создание с помощью существующего профиля
Служба ***createProfile*** также генерирует профиль с помощью существующего профиля и включает нижеследующие шаги:
a) служба ***requestExistingProfile*** запрашивает существующий профиль. Тип сервиса:
<service-type>="*requestExistingProfile*" с соответствующими атрибутами:
- profile_ID="the_profile_id"
b) служба ***returnExistingProfile*** получает существующий профиль. Тип сервиса:
<service-type>="*returnExistingProfile*" с соответствующими атрибутами:
- profile_content="the_profile_content"
- process_status="the_process_status"
c) служба ***processModifiedProfile*** запрашивает приемлемость модифицированного профиля. Тип сервиса:
<service-type> = “*processModifiedProfile*” с соответствующими атрибутами:
- profile_ID=”the_profile_id”
d) служба ***returnProcessingResult*** получает результат обработки. Тип сервиса:
<service-type> = *“returnProcessingResult*” с соответствующими атрибутами:
- ID_check_error=”ID_check_error”
- storage_error=”storage_error”

7.3.2 Доступ к профилю возможностей

7.3.2.1 Доступ через расширенный служебный интерфейс ESI
Служба ***accessProfile*** получает доступ к профилю через интерфейс ESI и включает нижеследующие шаги:
a) служба ***requestExistingProfile*** получает доступ к существующему профилю. Тип сервиса:
<service-type>="*requestExistingProfile*" с соответствующими атрибутами:
- profile_ID="the_profile_id"
b) служба ***returnExistingProfile*** получает запрошенный профиль. Тип сервиса:
<service-type>=”*returnExistingProfile*” с соответствующими атрибутами:
- profile_content=”the_profile_content”
- process_status=”the_process_status”
7.3.2.2 Доступ через блок программных средств организации производства MSU
Служба ***accessProfile*** также получает доступ к профилю через MSU и включает нижеследующие шаги:
a) служба ***requestExistingProfile*** получает доступ к существующему профилю. Тип сервиса:
<service-type>="*requestExistingProfile*"
b) служба ***returnExistingProfile*** получает запрошенный профиль. Тип сервиса:
<service-type>="*returnExistingProfile*" с соответствующими атрибутами:
- profile_content= "the_profile_content"
- process_status="the_process_status"

7.3.3 Модификация профиля возможностей

Служба ***modifyProfile*** модифицирует профиль и включает нижеследующие шаги:
a) служба ***requestExistingProfile*** получает доступ к существующему профилю. Тип сервиса:
<service-type>="*requestExistingProfile*" с соответствующими атрибутами:
- profile_ID="the_profile_id"
b) служба ***returnExistingProfile*** получает запрошенный профиль. Тип сервиса:
<service-type>=”*returnExistingProfile*” с соответствующими атрибутами:
- profile_content= “the_profile_content”

- process_status=”the_process_status”

c) служба ***processModifiedProfile*** запрашивает приемлемость модифицированного профиля. Тип сервиса:

<service-type>=”*processModifiedProfile*” с соответствующими атрибутами:

- profile_ID=”the_profile_id”

d) служба ***returnProcessingResult*** получает результат обработки. Тип сервиса:

<service-type>="*returnProcessingResult*" с соответствующими атрибутами:

- ID_check_error="ID_check_error"

- storage_error=”storage_error”

7.3.4 Проверка соответствия профиля возможностей

Служба ***validateProfile*** обеспечивает испытание существующего профиля и включает нижеследующие шаги:

a) служба ***requestExistingProfile*** получает доступ к существующему профилю. Тип сервиса:

<service-type>="*requestExistingProfile*" с соответствующими атрибутами:

- profile_ID="the_profile_id"

b) служба ***returnExistingProfile*** получает запрошенный профиль. Тип сервиса:

<service-type>="*returnExistingProfile*" с соответствующими атрибутами:

- profile_content="the_profile_content"

- process_status="the_process_status"

c) служба ***testProfile*** верифицирует незарегистрированный профиль. Тип сервиса:

<service-type>="*testProfile*"

d) служба ***returnTestResult*** получает результат испытаний и статус испытаний. Тип сервиса:

<service-type>="*returnTestResult*" с соответствующими атрибутами:

- test_result="the_test_result"

- test_status="the_test_status"

7.3.5 Стирание профиля возможностей

Служба ***deleteTemplate*** стирает существующий шаблон и включает нижеследующие шаги:

a) служба ***requestExistingProfile*** получает доступ к существующему профилю. Тип сервиса:

<service-type>="*requestExistingProfile*" с соответствующими атрибутами:

- profile_ID="the_profile_id"

b) служба ***returnExistingProfile*** получает запрошенный профиль. Тип сервиса:

<service-type>="*returnExistingProfile*" с соответствующими атрибутами:

- profile_content= "the_profile_content"

- process_status="the_process_status"

c) служба ***removeProfile*** удаляет профиль возможностей из архива. Тип сервиса:

<service-type>=”*removeProfile*”

d) служба ***returnRemoveResult*** получает статус удаления. Тип сервиса:

<service-type>="*returnRemoveResult*" с соответствующими атрибутами:

- remove_status="the_remove_status"

7.4 Служебные протоколы CCSI группы

7.4.1 Создание структуры класса возможностей

7.4.1.1 Создание с помощью формальной структуры

Служба createCCS генерирует новую структуру CCS с помощью формальной структуры и включает нижеследующие шаги:

a) служба ***requestBlankCCS*** запрашивает заготовку CCS. Тип сервиса:

<service-type>="*requestBlankCCS*"

b) служба ***returnBlankCCS*** получает заготовку CCS. Тип сервиса:

<service-type>="*returnBlankCCS*" с соответствующими атрибутами:

- CCS_content="the_CCS_content"

- access_status="the_access_status"

c) служба ***processFilliedCCS*** запрашивает приемлемость заполненной структуры CCS. Тип сервиса:

<service-type>="*processFilliedCCS*" с соответствующими атрибутами:

- CCS_ID="the_CCS_id"

d) служба ***returnProcessingResult*** получает результат обработки. Тип сервиса:
<**service-type**>="***returnProcessingResult***" с соответствующими атрибутами:
- ID_check_error="ID_check_error"
- storage_error="storage_error"

7.4.1.2 Создание с помощью существующей структуры CCS

Служба ***createCCS*** также генерирует шаблон с помощью существующей структуры CCS и включает нижеследующие шаги:

a) служба ***requestExistingCCS*** запрашивает существующую CCS. Тип сервиса:
<**service-type**>="requestExistingCCS" с соответствующими атрибутами:
- CCS_ID="the_CCS_id"

b) служба ***returnExistingCCS*** получает существующую CCS. Тип сервиса:
<**service-type**>="***returnExistingCCS***" с соответствующими атрибутами:
- CCS_content="the_CCS_content"
- process_status="the_process_status"

c) служба ***processModifiedCCS*** запрашивает приемлемость модифицированной CCS. Тип сервиса:
<**service-type**>="***processModifiedCCS***" с соответствующими атрибутами:
- CCS_ID="the_CCS_id"

d) служба ***returnProcessingResult*** получает результат обработки. Тип сервиса:
<**service-type**>="***returnProcessingResult***" с соответствующими атрибутами:
- ID_check_error="ID_check_error"
- storage_error="storag e_error"

7.4.2 Доступ к структуре класса возможностей

Служба accessCCS получает доступ к структуре CCS и включает нижеследующие шаги:

a) служба ***requestExistingCCS*** запрашивает существующую CCS. Тип сервиса:
<**service-type**>="***requestExistingCCS***" с соответствующими атрибутами:
- CCS_ID="the_CCS_id"

b) служба ***returnExistingCCS*** получает существующую CCS. Тип сервиса:
<**service-type**>="***returnExistingCCS***" с соответствующими атрибутами:
- CCS_content="the_CCS_content"
- process_status="the_process_status"

7.4.3 Модификация структуры класса возможностей

Служба modifyCCS модифицирует структуру CCS и включает нижеследующие шаги:

a) служба ***requestExistingCCS*** запрашивает существующую CCS. Тип сервиса:
<**service-type**>="***requestExistingCCS***" с соответствующими атрибутами:
- CCS_ID="the_CCS_id"

b) служба ***returnExistingCCS*** получает существующую CCS. Тип сервиса:
<**service-type**>="***returnExistingCCS***" с соответствующими атрибутами:
- CCS_content="the_CCS_content"
- process_status="the_process_status"

c) служба ***processModifiedCCS*** запрашивает приемлемость модифицированной CCS. Тип сервиса:
<**service-type**>="***processModifiedCCS***" с соответствующими атрибутами:
- CCS_ID="the_CCS_id"

d) служба ***returnProcessingResult*** получает результат обработки. Тип сервиса:
<**service-type**>="***returnProcessingResult***" с соответствующими атрибутами:
- ID_check_error="ID_check_error"
- storage_error=”storage_error”

7.4.4 Проверка соответствия структуры класса возможностей

Служба ***validateCCS*** обеспечивает испытание CCS и включает нижеследующие шаги:

a) служба ***requestExistingCCS*** запрашивает существующую CCS. Тип сервиса:
<**service-type**>="***requestExistingCCS***" с соответствующими атрибутами:
- CCS_ID="the_CCS_id"

b) служба ***returnExistingCCS*** получает существующую CCS. Тип сервиса:

<service-type>="***returnExistingCCS***" с соответствующими атрибутами:
- CCS_content="the_CCS_content"
- process_status="the_process_status"
c) служба ***testCCS*** верифицирует незарегистрированную CCS. Тип сервиса:
<service-type>="***testCCS***"
d) служба ***returnTestResult*** получает результат испытаний и статус испытаний. Тип сервиса:
<service-type>="***returnTestResult***" с соответствующими атрибутами:
- test_result ="the_test_result"
- test_status="the_test_status"

7.4.5 Стирание структуры класса возможностей

Служба ***deleteCCS*** стирает существующий шаблон и включает нижеследующие шаги:
a) служба ***requestExistingCCS*** запрашивает существующую CCS. Тип сервиса:
<service-type>="***requestExistingCCS***" с соответствующими атрибутами:
- CCS_ID="the_CCS_id"
b) служба ***returnExistingCCS*** получает существующую CCS. Тип сервиса:
<service-type>="***returnExistingCCS***" с соответствующими атрибутами:
- CCS_content="the_CCS_content"
- process_status="the_process_status"
c) служба ***removeCCS*** удаляет структуру CCS из архива. Тип сервиса:
<service-type>="***removeCCS***"
d) служба ***returnRemoveResult*** получает статус удаления. Тип сервиса:
<service-type>="***returnRemoveResult***" с соответствующими атрибутами:
- remove_status="the_remove_status"

7.5 Служебные протоколы расширенной группы обнаружения совпадений

Служба ***ExtendedMatcher*** сопоставляет профиль возможностей блока программ MSU с требуемым профилем, используя обнаружитель совпадений. Услуга включает нижеследующие шаги:
a) служба ***requestExistingProfile*** получает доступ к существующему профилю. Тип сервиса:
<service-type>="***requestExistingProfile***" с соответствующими атрибутами:
- profile_ID="the_profile_id"
b) служба ***returnExistingProfile*** получает запрошенный профиль. Тип сервиса:
<service-type>="***returnExistingProfile***" с соответствующими атрибутами:
- profile_content="the_profile_content"
- process_status="the_process_status"
c) служба ***requestMatching*** сопоставляет два доступных профиля. Тип сервиса:
<service-type>="***requestMatching***" с соответствующими атрибутами:
- profile_ID_1 ="the_profile_id_1"
- profile_ID_2="the_profile_id_2"
d) служба ***returnMatchingResult*** получает результат сопоставления. Тип сервиса:
<service-type>="***returnMatchingResult***" с соответствующими атрибутами:
- matching_level="the_matching_level"
- matching_report="the_matching_report"

8 Служба и протокол импорта словаря

8.1 Служба импорта словаря DictionaryImporting

Служба ***DictionaryImporting*** использует службу ***requestImportDictionary***, службу ***returnImportDictionary***, службу ***requestDictionary*** и службу ***returnDictionary***. Служба дает возможность пользователю импортировать библиотеку деталей в архив и просмотреть содержание архива (см. рисунок 19).

Служба ***DictionaryImporting*** включает нижеследующие шаги:

a) пользователь словаря инициирует службу ***requestImportDictionary*** объекта ***ImportServicePoint***, в котором параметром сервиса, ассоциированного со службой ***requestImportDictionary***, является идентификатор словаря;

b) поставщик сервисов инициирует службу ***returnImportResult*** объекта ***ImportServicePoint***, в котором параметрами службы ***returnImportResult*** являются результат импортирования и ошибка обработки;

c) пользователь словаря инициирует службу requestDictionary объекта ImportServicePoint, в котором параметром службы requestDictionary является идентификатор словаря;

d) поставщик сервисов инициирует службу returnDictionary объекта ImportServicePoint, в котором параметрами службы returnDictionary являются существующий словарь и ошибка обработки.

8.2 Протокол импорта словаря DictionaryImporting

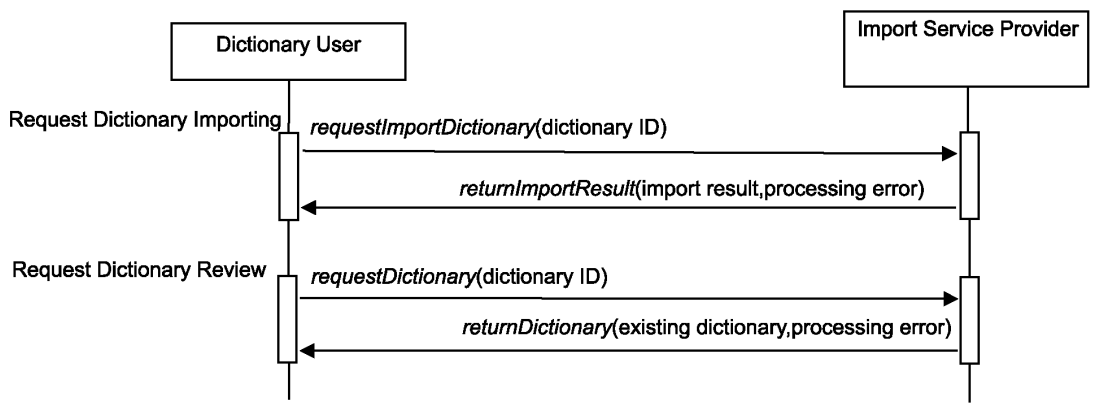
Служба ***DictionaryImporting*** импортирует словарь в архив и включает нижеследующие шаги:

a) служба ***requestImportDictionary*** запрашивает импортирование словаря. Тип сервиса:
<service-type>="*requestImportDictionary*" с соответствующими атрибутами:
- dictionary_ID="the_dictionary_id"

b) служба ***returnImportDictionary*** получает импортированный словарь. Тип сервиса:
<service-type>="returnImportingresult" с соответствующими атрибутами:
- importing_result="the_importing_result"
- process_status="the_process_status"

c) служба ***requestDictionary*** запрашивает словарь. Тип сервиса:
<service-type>="*requestDictionary*" с соответствующими атрибутами:
- dictionary_ID="the_dictionary_id"

d) служба ***returnDictionary*** получает запрошенный словарь. Тип сервиса:
<service-type>="*returnDictionary*" с соответствующими атрибутами:
- dictionary_content="the_dictionary_content"
- process_status="the_process_status"



Dictionary User	Пользователь словаря
Import Service Provider	Поставщик сервисов импортирования
Request Dictionary Importing	Запрос импртирования словаря
requestImportDistionary(dictionary ID)	***requestImportDictionary*** (идентификатор словаря)
returnImportResult (import result, processing error)	***returnImportingresult*** (результат импортирования, ошибка обработки)
Request Dictionary Review	Запрос пересмотра словаря
requestDictionary(dictionary ID)	***requestDictionary*** (идентификатор словаря)
returnDictionary(existing dictionary, processing error)	***returnDictionary*** (существующий словарь, ошибка обработки)

Рисунок 19 — Служба ***DictionaryImporting***

Приложение А
(справочное)

Модель возможностей, содержащая объекты данных MDD

А.1 Диаграмма модели возможностей

Производственная деятельность включает одну и более операций производственного процесса, ассоциированных с набором производственных функций в соответствии с ИСО 16100-1:2009, раздел 5.3. Для каждой операции можно разработать модель, содержащую объекты данных MDD, в соответствии с ИСО 15745-1.

Имеется однозначное соответствие между элементами дерева приложения и элементами дерева класса возможностей. Модель производственной деятельности отображается на модель класса возможностей, как показано на рисунке А.1.

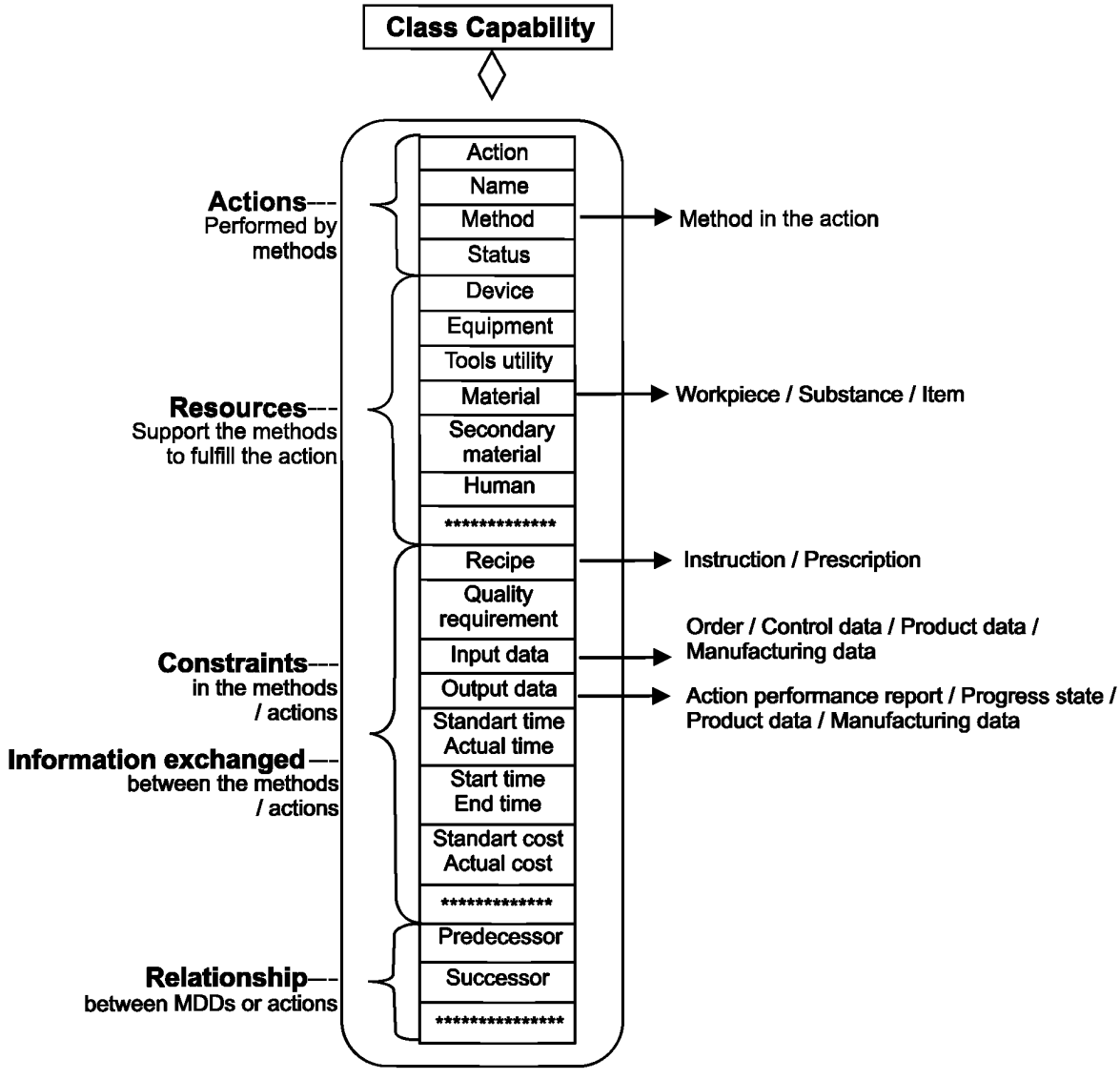
Рисунок А.1 описывает нижеследующие возможности структуры CCS в терминах объекта данных MDD:

a) операции производственной деятельности;

b) обмен ограничениями (информацией) между операциями;

c) ресурсы, использованные при выполнении операции;

d) соотношения между предшествующей и/или последующей операциями.



Class Capability	Возможности класса
Actions---Performed by methods	Операции (выполняются разными методами)
Action	Операция
Name	Название
Method	Метод
Status	Статус
Methods in the action	Метод выполнения операции
Device	Устройство
Equipment	Оборудование
Tools utility	Инструмент для выполнения операции
Material	Материал
Secondary material	Вспомогательный материал
Human	Человек
Resources---Support the methods to fulfill the action	Ресурсы (реализация метода выполнения операции)
Workpiece / Substance / Item	Заготовка/вещество/изделие
Recipe	Рецептура
Instruction / Prescription	Инструкция/Предписание
Quality requirement	Требование к качеству
Input data	Входные данные
Order / Control data / Product data / Manufacturing data	Приказ/данные управления/данные продукта/ производственные данные
Output data	Выходные данные
Action performance report / Process state / Product data / Manufacturing data	Отчет о выполнении операции/состояние развития/данные родукта/производственные данные
Standard time/Actual time	Стандартное время/фактическое время
Constraints---in the methods / actions	Ограничения (метода/операции)
Information exchanged---between the methods / actions	Обмен информации (между методами/операциями)
Start time/End time	Время начала/время окончания
Standard cost/Actual cost	Стандартные затраты/фактические затраты
Relationship---between MDDs or actions	Соотношения (между объектами данных MDD и операциями)
Predecessor	Предшествующая операция
Successor	Последующая операция

Рисунок А.1 — Отображение модели возможностей на модель производственной деятельности с помощью объектов данных MDD
Рисунок А.1, лист 2.

А.2 Синтаксис языка разметки XML для рассматриваемой модели возможностей

Ниже приведен пример синтаксиса языка разметки XML для модели возможностей, представленной в особой части шаблона.

```
<?xml version="1.O" encoding="UTF-8"?>
<xs:schema xmlns:xs="gost.e-ecolog.ru                    ">
<xs:element name="CapabilityProfiling">
    <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
        <xs:element name="type">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required"/>
            </xs:complexType>
        </xs:element>
        <xs:element name="CapabilityProfile">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="pkgtype">
                    <xs:complexType>
                        <xs:attribute name="version" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                    </xs:element>
                    <xs:element name="Common" type="CommonPartType"/>
                    <xs:element name="Specific" type="SpecificPartType"/>
```

```
                </xs:sequence>
                <xs:attribute name="date" type="xs:string" form="unqualified"/>
            </xs:complexType>
        </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="CommonPartType">
</xs:complexType>
<xs:complexType name="SpecificPartType">
<xs:sequence>
    <xs:element name="Reference_MDM_Name">
    <xs:complexType>
        <xs:attribute name="domain_Name" type="xs:string" form="unqualified"/>
    </xs:complexType>
    </xs:element>
    <xs :element. name="MDD_Description_Format" t.ype="MDD_Description" />
    <xs:complexType name="MDD_Description">
        <xs:sequence>
            <xs:choice>
                <xs:element name="Set_of_MDD_Objects">
                    <xs:complexType>
                        <xs:sequence minOccurs="0“ maxOccurs="unbounded">
                        <xs:element name="MDD_Name_action">
                            <xs :att.ribute name="name" t.ype="xs : string" use=" required" fixed=""/> <xs:attribute
                            name="method" type="xs:string" use="required" fixed=""/> <xs:attribute name="status"
                            type="xs:string" default=""/>
                        </xs:element>
                        <xs:element name="MDD_Name_exchanged_information">
                            <xs:complexType>
                                <xs:sequence min0ccurs="0" maxOccurs="unbounded">
                                <xs:choice>
                                    <xs:element name="information_out">
                                        <xs:complexType>
                                            <xs:attribute name="name" type="xs:string"form="unqualified"/>
                                        </xs:complexType>
                                        </xs:element>
                                        <xs:element name="information_in">
                                        <xs:complexType>
                                            <xs:attribute name="name" type="xs:string"form="unqualified"/>
                                         </xs:complexType>
                                        </xs:element>
                                    </xs:choice>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>          // end of exchanged_information
                        <xs : element name="MDD_Name_constraints" >
                        <xs:complexType>
                                <xs:sequence minOccurs="0" maxOccurs="unbounded">
                                 <xs:element name="Constraint_name">
                                        <xs:complexType>
                                            <xs:attribute name="name" type="xs:string"form="unqualified"/>
                                        </xs:complexType>
                                        </xs:element>
                                 < /xs: sequenc.e>
                            </xs:complexType>
                        </xs:elements              // end of constraints
                        <xs : element name="MDD_Name_resources" >
                            <xs:complexType>
                                <xs:sequence minOccurs="0" maxOccurs="unbounded">
                                        <xs:element name="Resource_name">
                                        <xs:complexType>
```

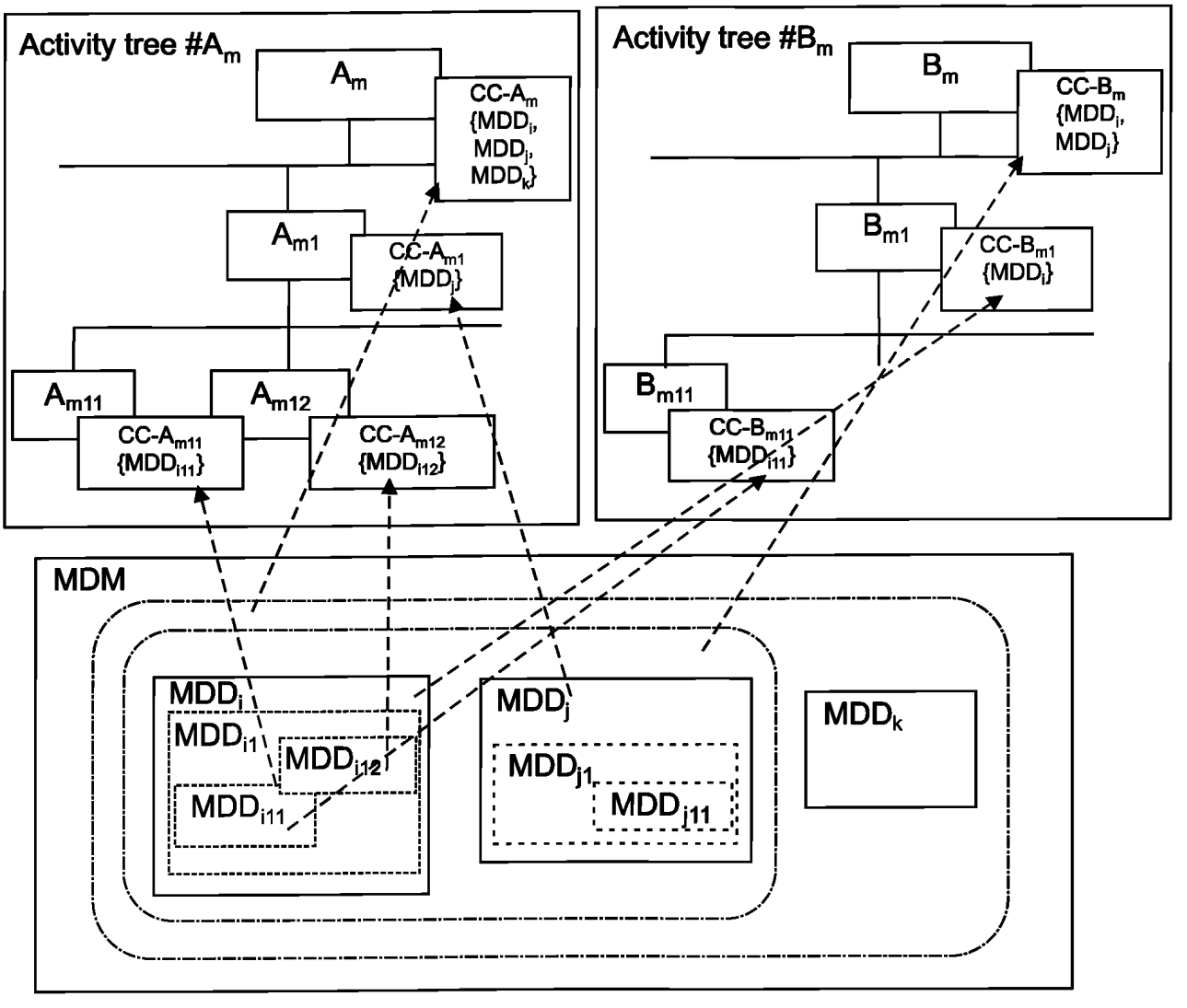
```
                                        <xs:attribute name="name" type="xs:string"form="unqualified" />
                                    </xs:complexType>
                                    </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                                </xs:element>          //  end  of  resources
                            </xs:sequence>
                        </xs:complexType>
                    </xs :element>                     //  end  of "Set_of_MDD_Objec.ts"
                    <xs:element name="List_of_MDD_Objects">
                    </xs:element>
                    <xs:element name="Time_ordered_MDD_Objects ">
                    </xs:element>
                    <xs: element name="Event_ordered_MDD_Objec.ts ">
                    </xs:element>
                </xs:choice>
                <xs : element name="List_of_lower_level " >
                    <xs:complexType>
                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                        <xs : element name=" Subac.tivity_name">
                            <xs: c.omplexType>
                                <xs:attribute name="name" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="subtemplate_name">
                            <xs:complexType>
                                <xs:attribute name="name" type="xs:string" form="unqualified"/> </
                            xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:sequence> // end for MDD_Description </xs:complexType>
    </xs: sequenc.e>
</xs:complexType> // end for SpecificPart </xs:schema>
```

А.3 Соотношения между объектами MDD и соответствующей моделью MDM

На рисунке А.2 сравниваются два дерева производственной деятельности (дерево А и дерево В) с различными структурами CCS. Каждый узел дерева имеет свой собственный класс возможностей и соответствующий шаблон профиля возможностей. Каждый шаблон использует объекты данных MDD для описания своих возможностей. Объекты MDD выбираются из одного множества объектов MDD для рассматриваемой модели MDM, например:

MDDi Є MDM|i Є [1..n]

Каждый объект MDD в рассматриваемой модели MDM должен быть определен. Он должен отличаться от других объектов. Уникальный идентификатор объекта MDD – отправная точка его семантического сопоставления.



Activity tree #A_m	Дерево производственной деятельности A

Примечание — Метки CC-A_m, CC-A_m1, CC-A_m11 и CC-A_m12 обозначают особые части класса возможностей узла производственной деятельности A_m, узла A_{m1}, узла A_{m11} и узла A_{m12}, соответственно. Аналогично, метки CC-B_m, CC-B_{m1} и CC-B_{m11} обозначают особые части класса возможностей узла производственной деятельности B_m, узла B_{m1} и узла B_{m11}, соответственно. Объекты MDD являются элементами указанных особых частей.

Рисунок А.2 — Соотношение между производственной деятельностью и соответствующими объектами данных MDD

Приложение В
(справочное)

Упрощенное сопоставление шаблонов профилей возможностей

В.1 Пример шаблона профиля возможностей

В.1.1 Пример 1

Ниже приведен пример синтаксиса языка разметки XML для шаблона профиля возможностей производственной деятельности типа A21 (“***getOperationMethod***”) по ИСО 16100-5, раздел В.2.

```
<?xml version="l.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="gost.e-ecolog.ru                    ">
    <xs:element name="CapabilityProfiling">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="Template">
                    <xs: c.omplexType>
                        <xs:attribute name="id" type="xs:string" use="required" fixed="A21"/>
                        <xs:attribute name="name" type="xs:string" use="required" fixed="getOperationMethod" />
                    </xs:complexType>
                </xs:element>
                <xs:element name="type">
                    <xs:complexType>
                        <xs:attribute name="id" type="xs:string"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="CapabilityProfile">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="pkgtype">
                            <xs:complexType>
                              <xs:attribute name="version" type="xs:string" form="unqualified"/> </xs:complexType>
                            </xs:element>
                            <xs:element name="Commcn" type="CommonPartType"/>
                            <xs:element. name="Specific" type= "SpecificPartType"/>
                        </xs:sequence>
                        <xs:attribute name="date" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                </xs:sequence>
        </xs:complexType>
  </xs:element>
  <xs:complexType name="CommonPartType">
     ..... ..
     ..... .
  </xs:complexType>
  <xs:complexType name="SpecificPartType">
        <xs:sequence»
            <xs:element name="Reference_MDM_Name*>
                <xs:complexType>
                    <xs:attribute name='domain_Name" type="xs:string" use="required" fixed="MESApplicationDomain"
  form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="format_name" type="MDD_Description"/>
        </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MDD_Description">
        <xs:sequence>
```

```
<xs:element name="Set_Of_MDD_Objects">
    <xs:complexType>
        <xs:sequence>
         <xs:element name="actionl">
         <xs:complexType>
              <xs:sequence maxOccurs="unbounded">
              <xs:element name="exchanged_information">
                   <xs:complexType>
                   <xs:sequence minOccurs="1" maxOccurs="1">
                        <xs:element name="information_out">
                             <xs:complexType>
                                  <xs:attribute name="name" use="required" type="xs:string" form="unqualified"
fixed="operation_type"/>
                             </xs:complexType>
                        </xs:elements </xs:sequences </xs:complexType>
              </xs:element>
              <xs:element name="Resources">
                        ..... ..
                        ..... .
              </xs:element>
              <xs:element name="Constraints">
                        ..... ..
                        ..... .
              </xs:elements </xs:sequences
              <xs:attribute name="name" type="xs:string" use="required" fixed="setOperationType"/>
              <xs:attribute name="method" type="xs:string" use="required" fixed="Set"/>
              <xs:attribute name="status" type="xs:string" default="mandatory"/>
              </xs:complexType>
              </xs:element>
         <xs:element name="action2"> <xs:complexType>
              <xs:sequence maxOccurs="unbounded">
              <xs:element name="exchanged_information">
              <xs:complexType>
                   <xs:sequence minOccurs="l" maxOccurs="l"> <xs:element name="information_in">
                   <xs:complexType>
                             <xs:attribute name="name" type="xs:string" use="required" form="unqualified"
fixed="operation_method(plan)"/>
                             </xs:complexType>
                             </xs:element>
                             </xs:sequence>
                             </xs:complexType>
                             </xs:element>
              <xs:element name="Resources">
                        ..... ..
                        ..... .
              </xs:element>
              <xs:element name="Constraints">
                        ..... ..
                        ..... .
              </xs:elements </xs:sequence>
              <xs:attribute name="name" type="xs:string" use="required" fixed="getOperationMethod"
form="unquali f ied"/>
              <xs:attribute name="method" type="xs:string" use="required" fixed="Get" form="unqualified"/>
              <xs:attribute name="status" type="xs:string" default="optional"/>
         </xs:complexTypes </xs:element>
         <xs:element name="action3">
         <xs:complexType>
              <xs:sequence maxOccurs="unbounded">
              <xs:element name="exchanged_information">
                   <xs:complexType>
                   <xs:sequence minOccurs="1" maxOccurs="1"> <xs:element name="information_in">
                   <xs:complexType>
```

```
                                        <xs:attribute name="name“ type="xs:string" use="required" form="unqualified"
fixed="recipe(plan)"/>
                                    </xs:complexType>
                                    </xs:element>
                                    </xs:sequence>
                                    </xs:complexType>
                                    </xs:element>
                                <xs:element name="Resources">
                                ..... ..
                                ..... .
                                </xs:element>
                                <xs:element name="Constraints">
                                ..... ..
                                ..... .
                        </xs:element>
                    </xs:sequence>
                    <xs :attribute name="name" type="xs:string" use="required" fixed="get.Recipe" form=“unqualified"/>
                    <xs:attribute name="method" type="xs:string" use="required" fixed="Get" form=“unqualified" />
                    <xs:attribute name="status" type="xs:string" default="optional"/>
                    </xs:complexType>
                </xs:element>
                </xs:sequence>
                </xs:complexType>
                </xs:element>
            <xs:element name="List_of_lower_level">
            .....
            .....
        </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

B.1.2 Пример 2

Ниже приведен пример синтаксиса языка разметки XML для шаблона профиля возможностей производственной деятельности типа B11 (“**receiveManufacturingInstruction**”) по ИСО 16100-5, раздел B.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="gost.e-ecolog.ru                                    ">
    <xs:element name="CapabilityProfiling">
      <xs: c.omplexType>
        <xs:sequences
            <xs:element name="Template">
              <xs:complexTypes
                <xs:attribute name="id" type="xs:string" use="required" fixed="BII"/>
                <xs:attribute name="name" type="xs:string" use="required" fixed="receiveManufacturingInstruetion" />
                  </xs:complexType>
            </xs:element>
            <xs:element name="type">
                <xs:complexType>
                    <xs:attribute name="id" type="xs:string"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="CapabilityProfile">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="pkgtype">
                            <xs:complexType>
                                <xs :attribut.e name="version" type="xs : string" form= "unqualified"/>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="Common" type="CommonPartType"/>
                        <xs:element name="Specific" type="SpecificPartType"/>
```

```
                    </xs:sequence>
                    <xs:attribute name="date" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:complexType name="CommonPartType">
    .....
    .....
</xs:complexType>
<xs:complexType name="SpecificPartType">
    <xs:sequence>
        <xs:element name="Reference_MDM_Name">
            <xs:complexType>
                <xs:attribute name="domain_Name" type="xs:string" use="required" fixed="MESApplicationDomain"
form="unqualified"/>
            </xs:complexType>
        </xs:element>
        <xs:element name='format_name" type="MDD_Description"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="MDD_Description">
    <xs:sequence>
    <xs:element name="Set_Of_MDD_Objects">
        <xs:complexType>
        <xs:sequence>
            <xs:element name="actionl">
            <xs:complexType>
                <xs:sequence maxOccurs="unbounded">
                <xs:element name="exchanged_information">
                    <xs:COmplexType>
                    <xs:sequence minOccurs="l" maxOccurs="l">
                        <xs:element name="information_in">
                            <xs:complexType>
                                <xs:attribute name="name" type="xs:string" use="required" form="unqualified"
fixed="product_order(plan)"/>
                            </xs:complexType>
                        </xs:element>
</xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="Resources">
...
...
                </xs:element>
                <xs:element name="Constraints">
                ...
...
                </xs:element>
                </xs:sequence>
                <xs:attri'bute name="name" type="xs:string" use="required" fixed="getProductOrder"/> <xs:attribute
                name="method" type="xs:string" use="required" fixed="Get"/>
                <xs:attribute name="status" type="xs:string" default="mandatory"/>
            </xs:complexType>
            </xs:element>
            <xs:element name="action2">
            <xs:complexType>
                <xs:sequence maxOccurs="unbounded">
                <xs:element name="exchanged_information">
                    <xs:complexType>
                    <xs:sequence minOccurs="1" maxOccurs="1">
```

```
<xs:element name="information_in">
                                        <xs:complexType>
                                            <xs:attribute name="name" type="xs:string" use="required" form="unqualified"
fixed="operating_instruction(plan)"/>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Resources">
                            .....
                            .....
                            </xs:element>
                            <xs:element name="Constraints">
                            .....
                            .....
                            </xs:element>
                            </xs:sequence>
                            <xs:attribute name="name" type="xs:string" use="required" fixed="getOperatingInstruction"
form="unqualified"/>
                            <xs:attribute name="method" type="xs:string" use="required" fixed="Get" form="unqualified"/>
                            <xs:attribute name="status" type="xs:string" default="optional"/>
                        </xs:complexType>
                        </xs:element>
                        <xs:element name="action3">
                        <xs:complexType>
                            <xs:sequence maxOccurs="unbounded">
                            <xs:element name="exchanged_information">
                                <xs:complexType>
                                <xs:sequence minOccurs="1" maxOccurs="l">
                                    <xs:element name="information_in">
                                        <xs:complexType>
                                    <xs:attribute name="name" type="xs:string" use="required"form="unqualified" fixed="item"/>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Resources">
                            .....
                            .....
                            </xs:element>
                            <xs:element name="Constraints">
                            .....
                            .....
                            </xs:element>
                            </xs:sequence>
                            <xs:attribute name="name" type="xs:string" use="required" fixed="getItem" form="unqualified"/>
                            <xs:attribute name="method" type="xs:string" use="required" fixed="Get" form="unqualified"/>
                            <xs:attribute name="status" type="xs:string" default="mandatory"/>
                        </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                    </xs:complexType>
                </xs:element>
                <xs:element name="List_of_lower_level">
                .....
                .....
                </xs:element>
                </xs:sequence>
        </xs:complexType>
        </xs:schema>
```

B.1.3 Пример 3

Ниже приведен пример синтаксиса языка разметки XML для шаблона профиля возможностей производственной деятельности типа A33 (“**monitorOperationCondition**») по ИСО 16100-5, раздел B.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="gost.e-ecolog.ru ">
    <xs:element name="CapabilityProfiling">
        <xs:ccmplexType>
            <xs:sequence>
                <xs:element name="Template">
                    <xs:complexType>
                    <xs:attribute name="id" type="xs:string" use="required" fixed="A33"/>
                    <xs:attribute name="name" type="xs:string" use="required" fixed="MonitorOperationConditicn" />
                    </xs:complexType>
                </xs:element>
                <xs:element name="type">
                    <xs:complexType>
                        <xs:attribute name="id“ type="xs:string"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="CapabilityProfile">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name=¯pkgtype">
                                <xs:complexType>
                                    <xs:attribute name="version" type="xs:string"form="unqualified"/>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="Common" type="CommonPartType"/>
                            <xs:element name="Specific" type="SpecificPartType"/>
                        </xs:sequence>
                        <xs:attribute name="date" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="CommonPartType">
    </xs:complexType>
    <xs:complexType name="SpecificPartType">
        <xs:sequence>
            <xs:element name="Reference_MDM_Name">
                <xs:complexType>
                    <xs:attribute name="domain_Name" type="xs:string" use="required"
fixed="MESApplicationDomain" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="format_name" type="MDD_Description"/>
        </xs: sequenc.e>
    </xs:complexType>
    <xs:complexType name="MDD_Description">
    <xs:sequence>
    <xs:element name= "Set_Of_MDD_Obj ects">
        <xs: c.omplexType>
        <xs: sequenc.e>
            <xs:element name="actionl">
            <xs:complexType>
                <xs:sequence maxOccurs="unbounded">
                <xs:element name="exchanged_information">
                    <xs: c.omplexType>
                    <xs:sequence minOccurs="l" maxOccurs="l">
                        <xs:element name="information_out">
```

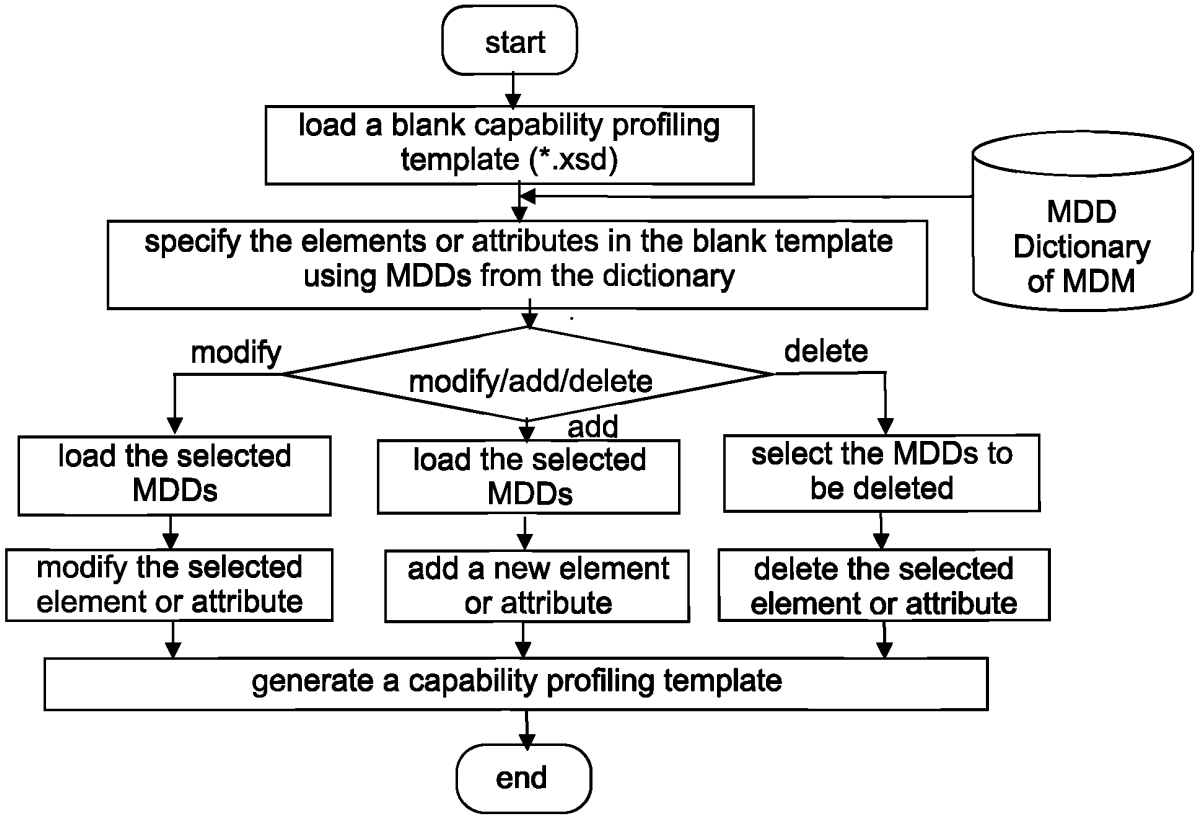
```
                                    <xs:complexType>
                                        <xsattribute name="name" type="xs:string" use="required" form="unqualified"
fixed="actual_equipment"/>
                                    </xs:complexType>
                                </xs:element>
                            </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Resources">
                        .....
                        .....
                        </xs:element>
                        <xs:element name="Constraints">
                        .....

                        .....
                        </xs:element>
                        < /xs: sequenc.e>
                        <xs:attribute name="name" type="xs:string" use="required" fixed="setActualEquipment"/>
                        <xs:attribute name="method" type="xs:string" use="required" fixed="Set"/>
                        <xs:attribute name="status" type="xs:string" default="mandatory"/>
                    </xs:complexType>
                    </xs:element>
                    <xs:element name="action2">
                    <xs:complexType>
                        <xs : sequence maxOcc.urs="unbounded">
                        <xs:element name="exchanged_information">
                            <xs:complexType>
                            <xs:sequence minOccurs="l" maxOc.curs="l">
                                <xs:element name="information_in">
                                    <xs:complexType>
                                        <xs:attribute name="name" type="xs:string" use="reauired" form="unqualified"
fixed="state(result)"/>
                                    </xs:complexType>
                                </xs:element>
                            </xs: sequenc.e>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Resources">
                        .....
                        .....
                        </xs:element>
                        <xs:element name="Constraints">
                        .....
                        </xs:element>
                        </xs:sequence>
                        <xs:attribute name="name" type="xs:string" use="required" fixed="getState" form="unqualified"/>
                        <xs:attribute name="method" type="xs:string" use="required" fixed="Get" form="unqualified"/>
                        <xs :attribute name=" status" type="xs:string" default="optional"/>
                    </xs:complexType>
                    </xs:element>
                </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="List_of_lower_level">
                <xs:complexType>
                <xs:sequence minOccurs="1" max0ccurs="4">
                    <xs:element name="subactivity_namel">
                    <xs:complexType>
                        <xs :attribute name="id" type="xs:string" use=" required" fixed="A331" form="unqualified"/>
                    </xs:complexType>
                    </xs:element>
```

```
            <xs:element name="subtemplate_namel">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required" fixed="A331" form="unqualified"/>
            </xs:complexType>
            </xs:element>
            <xs:element name="subactivity_name2">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required" fixed="A332" form="unqualified"/>
            </xs:complexType>
            </xs :element.>
            <xs:element name="subtemplate_name2">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required" fixed="A332" form="unqualified"/>
            </xs:complexType>
            </xs:element>
            <xs:element name="subactivity_name3">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required" fixed="A333" form="unqualified"/>
            </xs:complexType>
            </xs:element>
            <xs:element name="subtemplate_name3">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required" fixed="A333" form="unqualified"/>
            </xs:complexType>
            </xs:elements
            <xs:element name="subactivity_name4">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required" fixed="A334" form="unqualified"/>
            </xs:complexType>
            </xs:element>
            <xs:element name="subtemplat.e_name4">
            <xs:complexType>
                <xs:attribute name="id" type="xs:string" use="required" fixed="A334" form="unqualified"/>
            </xs:complexType>
            </xs:element>
        </xs:sequence>
        </xs:complexType>
    </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:schema>
```

B.2 Процедура создания шаблона профиля возможностей

Процедура создания нового шаблона профиля возможностей (с помощью формальной структуры шаблона профиля возможностей) показана на рисунке B.1.

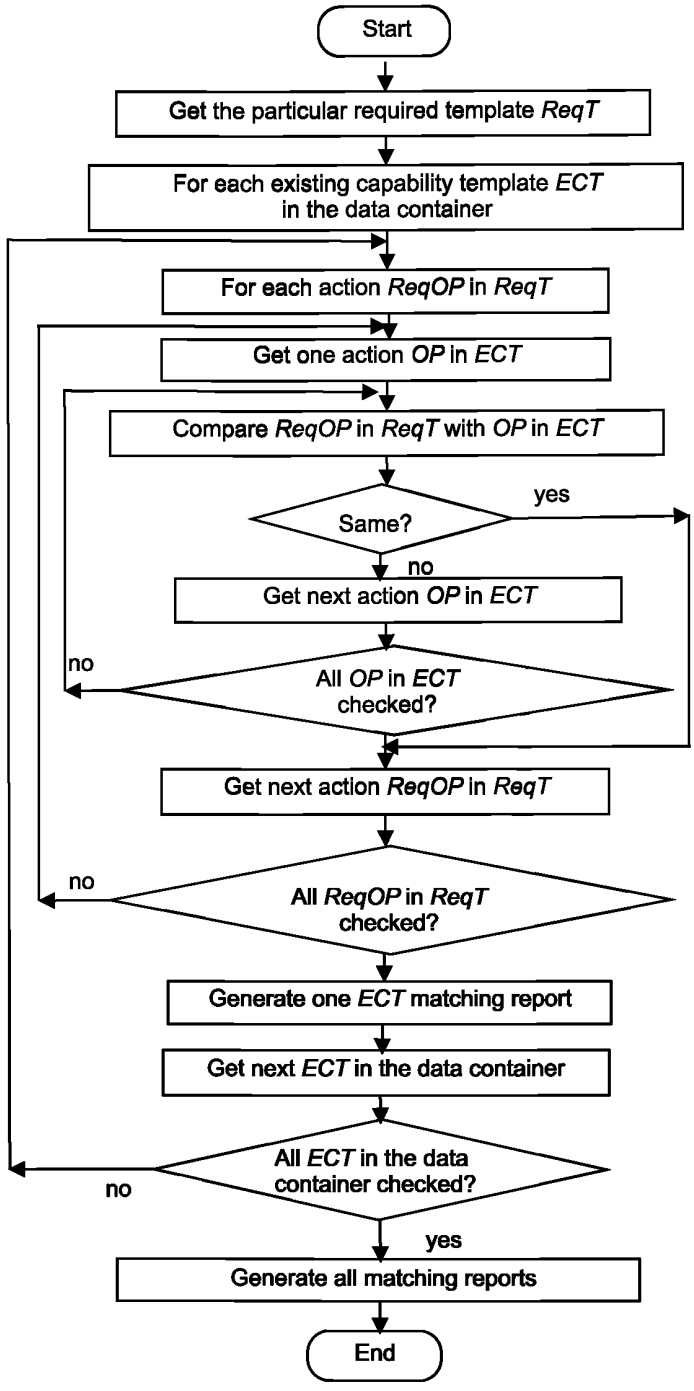


Start	Начало
load a bank capability profiling template (*.xsd)	Загрузка заготовки шаблона профилирования возможности
MDD Dictionary of MDM	Словарь объектов MDD для модели MDM
specify the elements or attributes in the blank template using MDDs from the dictionary	Задание элементов (атрибутов) заготовки шаблона, используя объекты MDD из словаря
Modify	Модифицировать
modify/add/delete	Модифицировать/добавить/стереть
Delete	Стирание
load the selected MDDs	Загрузка выбранных объектов MDD
select the MDDs to be deleted	Выбрать объекты MDD для стирания
modify the selected element or attribute	Модифицировать выбранный элемент (атрибут)
add a new element or attribute	Добавить новый элемент (атрибут)
delete the selected element or attribute	Стереть выбранный элемент (атрибут)
generate a capability profiling template	Создать шаблон профилирования возможностей
End	Конец

Рисунок B.1 — Процедура создания шаблона профиля возможностей

B.3 Процесс выбора надлежащего шаблона профиля возможностей

Процесс выбора надлежащего шаблона профиля возможностей из архива (в соответствии с требуемым шаблоном) показан на рисунке B.2.



Start	Начало
Get the particular required template ReqT	Получение требуемого шаблона
For each existing capability template ECT in the data container	Для каждого существующего шаблона возможностей в контейнере данных
For each action ReqOP in ReqT	Для каждой операции (требуемая операция в требуемом шаблоне)
Get one action OP in ECT	Выбрать одну операцию в существующем шаблоне
Compare ReqOP in ReqT with OP in ECT	Сравнить требуемую операцию в требуемом шаблоне с имеющейся операцией в существующем шаблоне
Same?	Одинаковы ли операции?

Yes	Да
no	Нет
Get next action OP in ECT	Перейти к следующей операции в существующем шаблоне
All OP in ECT checked?	Все ли операции существующего шаблона проверены?
Get next action ReqOP in ReqT	Перейти к следующей требуемой операции в требуемом шаблоне
All ReqOP in ReqT checked?	Все ли требуемые операции в требуемом шаблоне проверены?
Generate one ECT matching report	Создание отчета о результатах сопоставления с существующим шаблоном
Get next ECT in the data container	Перейти к следующему существующему шаблону в контейнере данных
All ECT in the data container checked?	Все ли существующие шаблоны в контейнере данных проверены?
Generate all matching reports	Создание отчета о результатах сопоставления
End	Конец

Обозначения на схеме:
ECT Существующий шаблон профиля возможностей
Req T Требуемый шаблон
ReqOP Требуемая операция
OP Операция

Рисунок В.2 — Процедура выбора шаблона профиля возможностей из Архива
Рисунок В.2, лист 2.

Приложение С
(справочное)

Профили, полученные с помощью шаблона профиля возможностей

С.1 Профиль “getOperationMethod”

Ниже приведен пример синтаксиса языка разметки XML для профиля производственной деятельности типа A21 (“**getOperationMethod**”) по ИСО 16100-5:2009, раздел B.2.

```
<?xml version="1.O" encoding="utf-8" ?>
<CapabilityProfiling xmlns:xsi=gost.e-ecolog.ru xsi:noNamespaceSchemaLocation="D:\
newDB\MonitorOperationCondition.xsd"> <Template id="A21" name="getOperationl>lethod" />
  <t.ype id="MSU profile" />
  <CapabilityProfi1e>
      <pkgtype version="1.1.1" />
      <Common>
      <MSU_Capabi1ity>
            <ID>getOperationMethod</ID>
      </MSU_Capabi1ity>
      <ReferenceCapabilityClassStruct.ure id="MESTreeA" /> <Capabilit.y_Class_Name name="get,Operat.ionMethod"
      /> <Reference_Capability_Class_Structure_Name name="MESTreeA" />
      </Common>
      <Specific>
      <Reference_MDM_Name domain_Name="MESApplicationDomain" />
      <format_name>
            <Set_Of_MDD_Objects>
            <action1 name="setOperationType" method="Set" status="mandatory">
            <exchanged_information>
            <information_out_in>
                <information_out name="operation_type"/>
            < / information_out_in>
            </exchanged_information>
            <Resources/>
            <Constraints/>
            </actionl>
            <action2 name="getOperationMethod" method="Get." status="optional "> <exchanged_information>
            <information_out_in>
                <information_in name="operation_method(plan)" />
            </information_out_in>
            </exchanged_information>
            <Resources />
            <Constraints />
            </action2>
            <action3 name="get.Recipe" method="Get" status=" optional" > <exchanged_information>
            <information_out_in>
                <information_in name="recipe(plan)" />
            </information_out_in>
            </exchanged_information>
            <Resources />
            <Constraints />
            </action3>
            </Set_Of_MDD_Objects>
            <List_of_lower_level />
      </format_name>
      </Specific>
  </CapabilityProfile>
  </Capabi1ityProfi1ing>
```

С.2 Профиль "receiveManufacturingInstruction"

Ниже приведен пример синтаксиса языка разметки XML для профиля производственной деятельности типа B11 ("**receiveManufacturingInstruction**") по ИСО 16100-5:2009, раздел В.3.

```
<?xml version="l.0" encoding="utf-8" ?>
<Capabi1ityProfi1ing xmlns:xsi =gost.e-ecolog.ru
xsi:noNamespaceSchemaLocation="D:\newDB\MonitorOperationCondition.xsd">
  <Template id="Bll" name="receiveManufacturingInstruction" />
  <type id="MSU profile" />
  <CapabilityProfile>
      <pkgtype version="1.1.1" />
      <Common>
      <MSU_Capabi1i ty>
            <ID>receiveManufacturingInstruction</ID>
      </MSU_Capabi1i ty>
      <ReferenceCapabilityClassStructure id="MESTreeB" />
      <Capabilit.y_Class_Name name="receiveManufacturingInstruetion" />
      <Reference_Capability_Class_Structure_Name name="MESTreeB" />
      </Common>
      <Specific>
      <Reference_MDM_Name domain_Name="MESApplicationDomain" />
      <format_name>
            <Set_Of_MDD_Objects>
            <action1 name="getProductOrder" method="Get" status="mandatory">
            <exchanged_information>
            <information_out_in>
                <information_in name="product_order(plan)" />
            </information_out_in>
            </exchanged_information>
            <Resources />
            <Constraints />
            </actionl>
            <action2 name="getOperatingInstruction" method="Get" status="optional">
            <exchanged_information>
            <information_out_in>
                <information_in name="operat.ing_instruction(plan) " />
            </information_out_in>
            </exchanged_information>
            <Resources />
            <Constraints />
            </action2>
            <action3 name="get.Item" method="Get" status=" optional">
            <exchanged_information>
            <information_out_in>
                <information_in name="item"/>
                </information_out_in>
            </exchanged_information>
            <Resources />
            <Constraints />
            </action3>
            </Set_Of_MDD_Objects>
            <List_of_lower_level />
      </format_name>
      </Specific>
  </Capabi1ityProfile>
  </Capabi1ityProfi1ing>
```

С.3 Профиль "monitorOperationCondition"

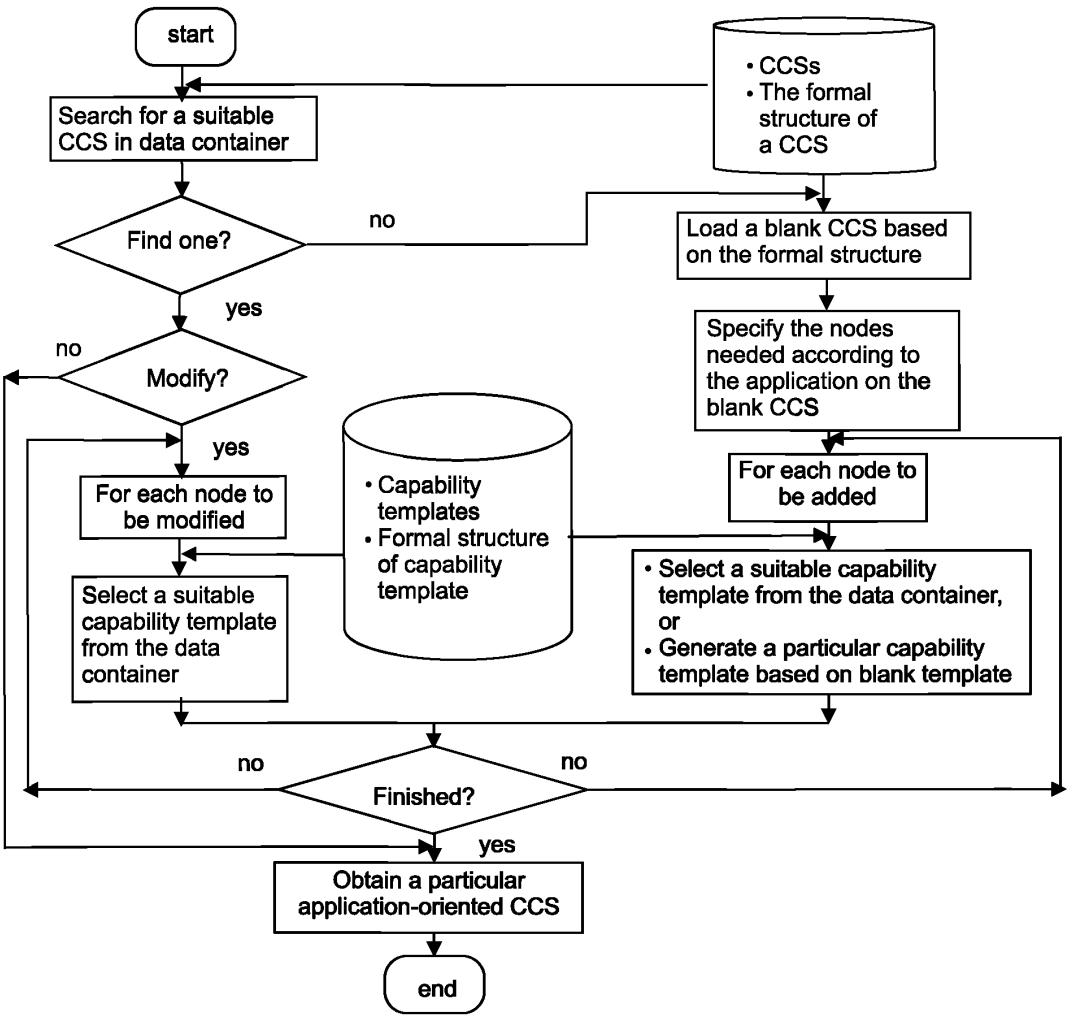
Ниже приведен пример синтаксиса языка разметки XML для профиля производственной деятельности типа A33 ("**monitorOperationCondition**") в ИСО 16100-5:2009, раздел В.2.

```
 <?xml version=" 1.0" encoding="ut.f-8" ?>
<CapabilityProfiling xmlns:xsi=gost.e-ecolog.ru
xsi :noNamespaceSchemaLocation="D: \newDB\MonitorOperationCondition.xsd" >
<Template id="A33" name="MonitorOperationCondition" />
    <type id="MSU profile" />
    <CapabilityProfile>
    <pkgtype version="1.1.1" />
    <Common>
    <MSU_Capabi1i ty>
        <ID>MSUMoni torOperationCondi tion</XD>
    < / I<ISU_Capabi1ity>
    <ReferenceCapabilityClassStructure id="MESTreeA" />
    <Capability_Class_Name name="MonitorOperationCondition" />
    <Reference_Capability_Class_Structure_Name name="MESTreeA" />
    </Common>
    <Specific>
    <Reference_MDM_Name domain_Name="MESApplicationDomain" />
    <format_name>
        <Set_Of_MDD_Objects>
          <actionl name="setMonitorCondition" method="Set" status="mandatory">
          < exchanged_information>
              <information_out_in>
              <information_out name="actual equipmentl" />
              </information_out_in>
          </exchanged_information>
          <Resources />
          <Constraints />
          </actionl>
          <action2 name="getMonitorCondition" method="Get" st.atus="opt.ional" >
          <exchanged_information>
              <information_out_in>
              <information_in name="statel" />
              </information_out_in>
          </exchanged_information>
              <Resources />
              <Constraints />
          </action2>
          < / Set_Of_MDD_Objects>
          <List_of_lower_level>
          <subac.tivity_namel id="A331" />
          <subtemplate_namel id="A331" />
          <subactivity_name2 id="A332" />
          <subtempiate_name2 id="A332" />
          <subactivity_name3 id="A333" />
          <subtemplate_name3 id="A333" />
          <subactivity_name4 id="A334" />
          <subtemplate_name4 id="A334" />
          </List_of_lower_level>
    </format_name>
    </Specific>
 </CapabilityProfi1e>
 </Capabi1ityProfi1ing>
```

Приложение D
(справочное)

Процедура создания структуры класса возможностей

Процедура создания структуры класса CCS показана на рисунке D.1.



start	Начало
Seach for a suitable CCS in data container	Поиск необходимой структуры в контейнере данных
- CCSs - The formal structure of a CCS	- структуры CCS - формальная структура CCS
Find one?	Удалось ли найти?
yes	Да
no	Нет
Load a blank CCS based on the formal structure	Загрузка заготовки CCS, основанной на использовании формальной структуры
Modify?	Модифицировать?

Specify the nodes needed according to the application on the blank CCS	Указание узлов, соответствующих приложению заготовки CCS
For each node to be modified	Для каждого модифицируемого узла
Select a suitable capability template from the data container	Выбор подходящего шаблона возможностей из контейнера данных
- Capability templates - Formal structure of capability template	- шаблоны возможностей - формальная структура шаблона возможностей
For each node to be added	Для каждого добавляемого узла
- Select a suitable capability template from the data container, or - Generate a particular capability template based on blank template	- выбор подходящего шаблона возможностей из котейнера данных - создание особого шаблона возможностей на основе заготовки
Finished?	Закончить работу?
Obtain a particular application-oriented CCS	Получение особой структуры CCS для рассматриваемого приложения
end	Конец

Рисунок D.1 — Процедура создания структуры класса CCS

Рисунок D.1, лист 2

Приложение Е
(справочное)

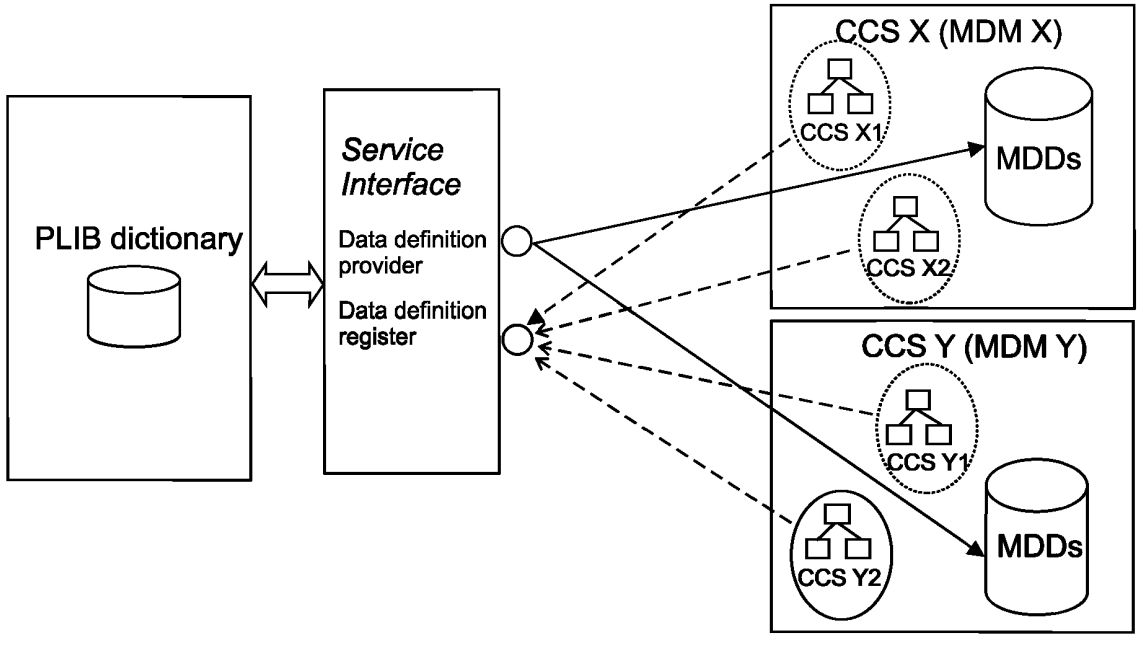
Отображение библиотеки деталей PLIB на объекты данных MDD

Е.1 Преимущества отображения

Словарь деталей PLIB (представляемый в электронной форме) включает определения описаний семейств деталей (классов PLIB) и атрибутов (определенных как «свойства» в ИСО 13584). Настоящая информация представляется кодом, известным как Базовая семантическая единица (BSU).

К записи BSU добавляются метки мульти-языка. Они поддерживают мульти-язычные определения. С помощью указанного механизма, профиль возможностей получает различные названия атрибутов (названия свойств в библиотеке деталей PLIB) и определения. В результате, механизм сопоставления просто оказывается электронным кодом, так как элементы BSU также представляются в электронной форме. Преимущество заключается в том, что данное сравнение проще, чем сравнение элементов BSU и объектов языка XML.

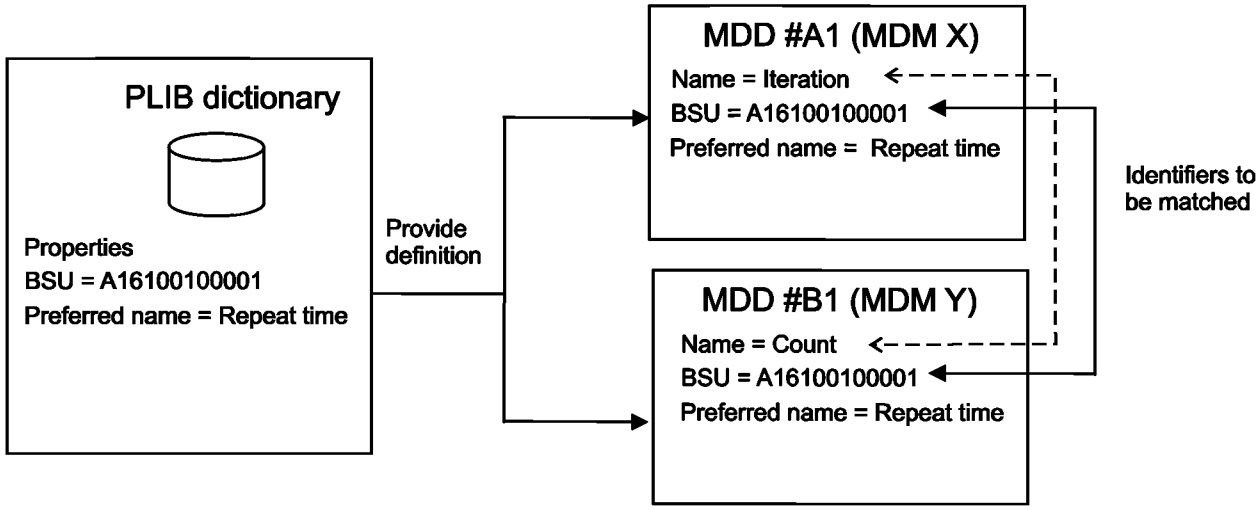
На рисунке Е.1 показано соотношение между элементами библиотеки PLIB и объектами данных MDD.



PLIB dictionary	Библиотека деталей
Service Interface	Интерфейс услуг
Data definition provider	Поставщик определений данных
Data definition register	Реестр определений данных

Рисунок Е.1 — Соотношение между элементами библиотеки PLIB и объектами данных MDD через интерфейс услуг

На рисунке Е.2 показаны два объекта MDD из одного словаря PLIB. Указанные объекты MDD имеют одинаковые коды BSU, но их названия различны. Механизм сравнения устанавливает, что указанные объекты MDD одинаковы, так как одинаковы их коды BSU.



PLIB dictionary	Словарь деталей
Properties	Свойства
Preferred name = Repeat time	Предпочтительное имя = время повторения
Name = Iteration	Имя = Итерация
Name = Count	Имя = Счет
Provide definition	Получение определения
Identifiers to be matched	Сопоставляемые идентификаторы

Рисунок Е.2 — Сравнение объектов MDD, используя идентификаторы понятий PLIB

Е.2 Пример отображения

Процесс создания заданного шаблона профиля возможностей начинается с рассмотрения формальной структуры MDD шаблона, описанной в ИСО 16100-5:2009, раздел 6.5.2. Рассматриваемый MDD шаблон создается после того, как выполнено отображение подмножества объектов MDD (из библиотеки PLIB) на формальную структуру MDD.

Ниже приведен пример применения схемы языка XML для рассматриваемого шаблона MDD.

```
<?xml version="1.O" encoding="UTF-8"?>
<xs : schema xmlns :xs="http: //gost.e-ecolog.ru                MLSchema">
    <xs:element name="MDD">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="MDD_Name">
                    <xs:complexType>
                        <xs:attribute name="name" type="xs:string" form="process order"/>
                        <xs:attribute name="DICTIOIS!ARY_ID" type= "xs : string" form=" 9999/IS016100-
XXX"/>
                        <xs:attribute name="PARENT" type="xs:string" form="F16100002E"/>
                        <xs:attribute name="BSU" type="xs:string" form="F16100001C"/>
                        <xs:attribute name="PREFERRED_NAME" language="en" type="xs:string"
form="process order"/>
                        <xs:attribute name="VERSION" type="xs:string" form="001" />
                        <xs:attribute name="REVISION" type="xs:string" form="001" />
                    </xs:complexType>
                </xs:element>
                <xs : element name= "Reference_MDM_Name">
                    <xs:complexType>
                        <xs:attribute name="name" type="xs:string" form="unqualified"/>
                    </xs:complexType>
                </xs:element>
                <xs:element name="List_Of_Attributes">
                    <xs:complexType>
```

```
                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                        <xs:element name="Attribute">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Attribute_Name">
                                        <xs:complexType>
                                            <xs:attribute name="name" type="xs:string"
form="Source"/>
                                            <xs:attribute name="DICTIONARY_ID" type="xs:string"
form="9999/IS016100-XXX"/>
                                            <xs:attribute name="PARENT" type="xs:string"
form="F16100001"/>
                                            <xs:attribute name="BSU" type="xs:string"
form="A161000001"/>
                                            <xs :attribute name="PREFERRED_NAME" language="en"
type="xs:string" form="Source"/>
                                            <xs:attribute name="PREFERRED_NAME" language="ja"
type="xs:string" form="Japanese-Source-Name"/>
                                            <xs:attribute name="VERSION" type="xs:string"
form="001" />
                                            <xs:attribute name="REVISION" type="xs:string"
form="001" />
                                        </xs:complexTypes
                                        </xs:element>
                                    <xs:element name="Attribute_Type">
                                        <xs:complexType>
                                            <xs:attribute name="type" type="xs:string"
form="unqualified"/>
                                        </xs:complexTypes
                                        </xs:element>
                                </xs:sequence>
                                <xs:attribute name="id" type="xs:string" form="unqualified"/>
                            </xs:complexType>
                        </xs:element>
                        <xs:element name="Attribute">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Attribute_Name">
                                        <xs:complexType>
                                            <xs:attribute name="name" type="xs:string"
form="Destination"/>
                                            <xs:attribute name="DICTIONARY_ID" type="xs:string"
form="9999/IS016100-XXX"/>
                                            <xs:attribute name="PARENT" type="xs:string"
form="F16100001"/>
                                            <xs:attribute name="BSU" type="xs:string"
form="A161000002"/>
                                            <xs:attribute name="PREFERRED_NAME" language="en"
type="xs:string" form="Destination"/>
                                            <xs :attribute name="PR.EFERRED_NAME" language= " ja"
type="xs:string" form="Japanese-Destination-Name"/>
                                            <xs:attribute name="VERSION" type="xs:string"
form="001" />
                                            <xs:attribute name="REVISION" type="xs:string"
form="001" />
                                        </xs:complexType>
                                    </xs:elements
                                    <xs:element name="Attribute_Type">
                                        <xs:complexType>
                                            <xs:attribute name="type" type="xs:string"
form="unqualified"/>
                                        </xs: complexType>
```

```
                    </xs:elements
                    </xs:sequences
                <xs:attribute name="id" type="xs:string" form="unqualified"/>
                </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Приложение F
(справочное)

Отображение открытого словаря OTD на объекты данных MDD

F.1 Преимущества отображения

Открытый технический словарь (OTD) – это (представленный в электронной форме) сборник записей различных типов понятий, включая понятия класса, свойства, управляемого значения свойства, единицы измерений, определителей мер и валют. Каждая запись включает, как минимум, уникальный идентификатор, термин и определение. Также имеется справочник идентификаторов, используемый для указания (в рассматриваемой области) минимального набора свойств, требуемых для описания как элементов класса, так и ограничений (соотношений) свойств.

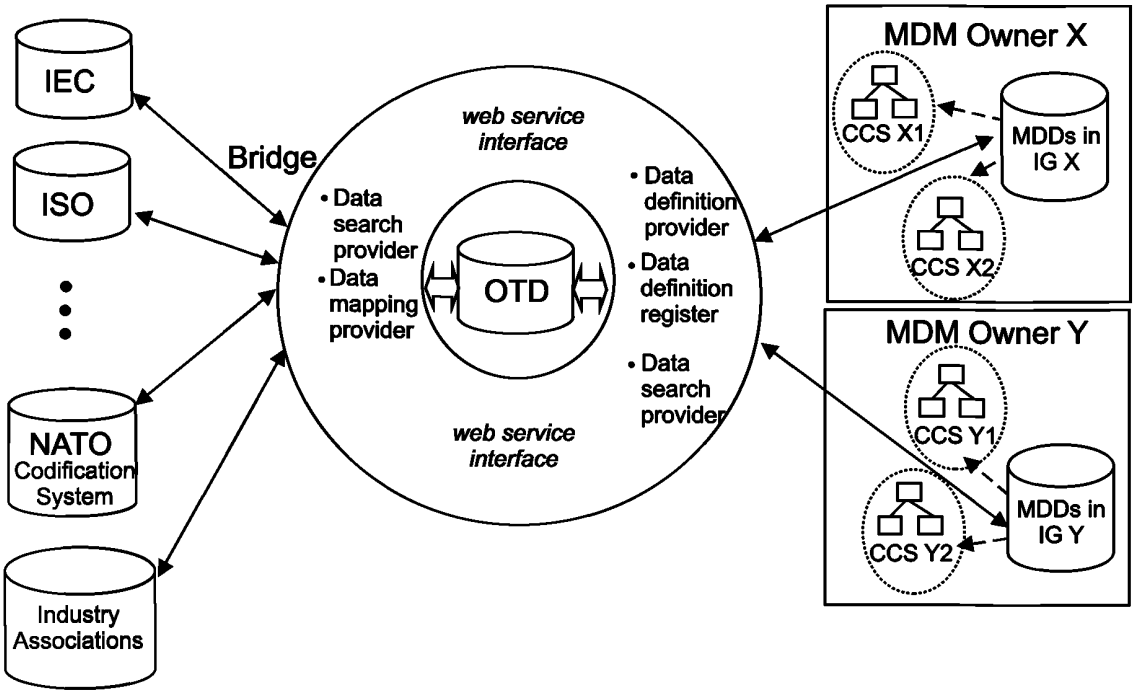
В соответствии с ИСО 22745, продукт описывается классом, которому он принадлежит, и набором свойств (пар значений). По описанию изделия, соответствующему ИСО 22745, каждое понятие представляется соответствующим уникальным идентификатором, содержащемся в словаре OTD. ИСО 22745 занимает нейтральную позицию в рассматриваемой классификации.

Соответствующее ИСО 16100 описание MDD, содержащее информацию о производственной деятельности, ресурсах, возможностях программного обеспечения и соотношениях между объектами MDD, может быть закодировано как описание продукта, соответствующего ИСО 22745, с помощью понятий словаря OTD. Возможности MSU или возможности, определяемые приложением, также могут быть выражены в терминах объектов MDD с помощью словаря OTD. Описание продукта, удовлетворяющее требованиям ИСО 22745, может быть связано с эквивалентными описаниями продукта, данными как в ИСО 22745, так и в других стандартах (например, ИСО 13584).

Для сопоставления двух объектов MDD (с помощью понятий словаря OTD) производится сравнение однозначных идентификаторов понятий. Если два различных идентификатора понятий OTD представляют одно и то же понятие, то настоящий факт следует отразить в словаре OTD как соотношение эквивалентности понятий. Указанные идентификаторы либо указываются явно среди объектов MDD, либо выделяются с помощью накладываемых связей.

Рисунок F.1 дает соотношение между объектами MDD и открытым словарем OTD.

Рисунок F.2 показывает два объекта MDD (обозначение #A1 относится к объекту MDD класса возможностей A1; обозначение #B1 относится к объекту MDD класса возможностей B1), которые ссылаются на один глобальный идентификатор продукта OTD и на одно предпочтительное имя. Два объекта MDD с различными названиями в соответствующих словарях представлены соответствующими терминами OTD в ресурсе OTD с разделенным доступом (например, в сервере). Общий глобальный идентификатор является мостом между индивидуальными словарями для сопоставления объектов MDD.



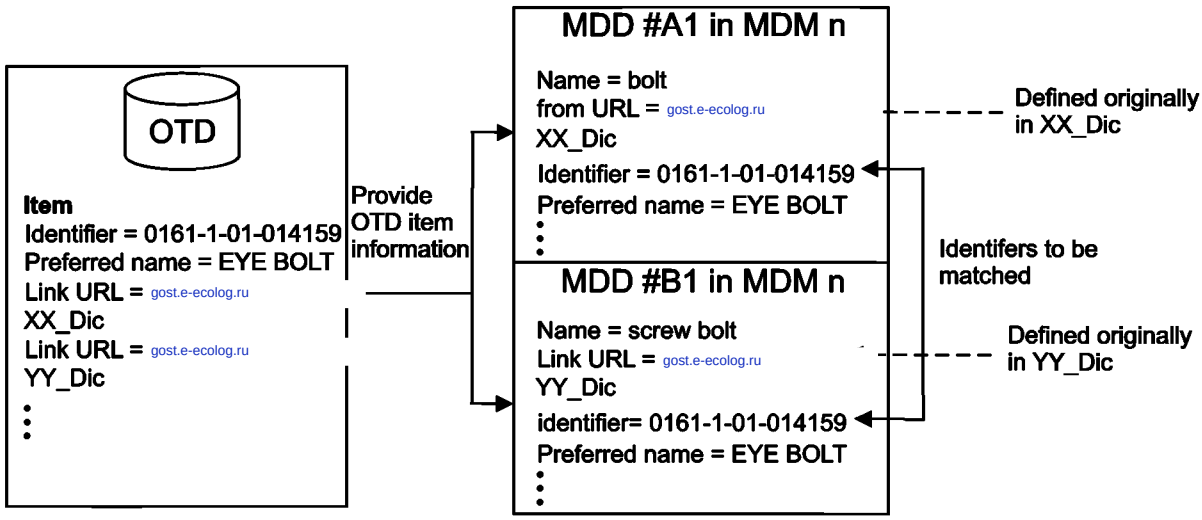
NATO Codification System	Система кодификации НАТО
Bridge	Мост
MDM Owner X	Собственник модели X
MDDs in IG X	Объекты данных MDD в справочнике идентификаторов X
web service interface	Интерфейс сетевых услуг
Data search provider	Провайдер поиска данных
Data mapping provider	Провайдер отображения данных
Data definition provider	Поставщик определений данных
Data definition register	Реестр определений данных
OTD	Открытый технический словарь
Industry Associations	Промышленные ассоциации

IG Справочник идентификаторов (см. ИСО 22745-2)

 Подмножество объектов MDD, используемое в структуре CCS

 Двусторонний обмен информацией

Рисунок F.1 — Соотношение между открытым словарем OTD и объектом данных MDD, устанавливаемое интерфейсом услуг



OTD	Открытый технический словарь
Item	Изделие
Identifier	Идентификатор
Preferred name = EYE BOLT	Предпочтительное имя = рым-болт
Link URL =	Рабочий язык ресурса =
MDD #A1 in MDM n	Объект MDD класса A1 в модели MDM
Name = bolt	Имя = болт
Provide OTD item information	Доставляет информацию об изделии из словаря OTD
from URL	На языке ресурса
Name = screw bolt	Имя = болт с головкой
Defined originally in XX_Dic	Изначально определен в словаре XX
XX_Dic	Словарь XX
Identifiers to be matched	Сопоставляемые идентификаторы
Defined originally in YY_Dic	Изначально определен в словаре YY

Рисунок F.2 — Сравнение объектов MDD и идентификаторов понятий с помощью открытого технического словаря OTD

F.2 Пример отображения

Как и в разделе E.2 (по аналогии с формальным шаблоном структуры MDD, описанной в ИСО 16100-5:2009, раздел 6.5.2) рассматриваемый пример шаблона профиля возможностей начинается с создания шаблона MDD, используя подмножество объектов MDD (на базе понятий OTD), выраженных в терминах элементов, определенных в словаре OTD для рассматриваемой модели MDM.

Ниже на языке XML приведена схема рассматриваемого шаблона MDD.

```
<?xml version="1.O" encoding="UTF-8"?>
<xs:schema xmlns:xs="gost.e-ecolog.ru                    ">
    <xs:element name="MDD">
        <xs:complexType>
            <xs:sequence>
                <xs:element name="MDD_Name">
                    <xs:complexType>
                        <xs:attribute name="name" type="xs:string" form="qualified"
                        fixed="process operation"/>
                        <xsrattribute name="GlobalItemIdentifier" type="xs:string" form="qualified"
                        fixed="process_MDM_IGxxx"/>
                        <xs:attribute name="GlobalItemCategory" type="xs:string" form="qualified"
                        fixed=" process_MDM_IGxxx "/>
                        <xs:attribute name="Title" type="xs:string" form="qualified"
                        fixed="process_MDM_IGxxx "/>
                        <xsrattribute name="Definition" type="xs:string" form="qualified"
                        fixed="process_MDM_IGxxx "/>
                        <xs:attribute name="DateAdded" type="xs:string" form="qualified"
                        fixed="process_MDM_IGxxx "/>
```

```
                    <xs:attribute name="SecretariatReference" type="xs:string" form="qualified"
                fixed="process_MDM_IGxxx"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="Reference_MDM_Name">
                <xs:complexType>
                    <xs:attribute name="name" type="xs:string" form="unqualified"/>
                </xs:complexType>
            </xs:element>
            <xs:element name="List_Of_Attributes">
                <xs:complexType>
                    <xs:sequence minOccurs="0" maxOccurs="unbounded">
                        <xs:element name="Attribute">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element name="Attribute_Name">
                                        <xs:complexType>
                                            <xs:attribute name="AttributeTitle"
                                        type="xs:string" form="qualified" fixed="
                                        process_MDM_IGxxx "/>
                                            <xs:attribute name="AttributeDefinition"
                                        type="xs:string" form="qualified" fixed="
                                        process_MDM_IGxxx"/>
                                        </xs:complexType>
                                    </xs:element>
                                    <xs:element name="Attribute_Type">
                                        <xs:complexType>
                                        <xs:attribute name="GlobalAttributeCategory"
                                    type="xs:string" form="qualified"
                                    fixed="process_MDM_IGxxx"/>
                                        <xs:attribute name="AttributeDateAdded"
                                    type="xs:string" form="qualified" fixed="process_MDM_IGxxx"/>
                                        <xs:attribute name="AttributeSecretariatReference"
                                    type='xs:string' form='qualified" fixed=" process_HDM_IGxxx"/>
                                        </xs:complexType>
                                    </xs:element>
                                </xs:sequence>
                                <xs :attribute name="GlobalAttributeXdentifier" type="xs:string"
                            form="qualified" fixed="process_MDM_IGxxx"/>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
```

Приложение G
(справочное)

Процедура сопоставления двух профилей

Комплект услуг расширенной группы обнаружения совпадений может быть использован для сопоставления профилей возможностей с помощью нескольких структур CCS. Целью сопоставления двух профилей является отбор подходящих элементов MSU из архива в соответствии с требуемым профилем возможностей для данного вида производственной деятельности в рассматриваемом производственном приложении.

«Процедура сопоставления профилей возможностей» определена в ИСО 16100-5:2009, рисунок 12. Ниже приведены 5 шагов данной процедуры:

- шаг 1: сравнение двух словарных идентификаторов двух профилей;
- шаг 2: сравнение названий двух моделей MDM;
- шаг 3: сравнение двух форматов определений;
- шаг 4: преобразование объектов MDD в соответствующие определения;
- шаг 5: последовательное сравнение определений возможностей в требуемом профиле возможностей с соответствующими определениями в профиле возможностей MSU.

На каждом шаге, уникальный идентификатор объекта MDD (уникальный внутри одной модели MDM) является начальной точкой семантического сравнения объектов MDD. Рассматривамая процедура "сравнения определений возможностей" (шаг 5) фокусируется на сравнении описаний "**MDD_Description**". Процедура сравнения (например, списка объектов "**List_of_MDD_Objects**") показана на рисунке G.1.

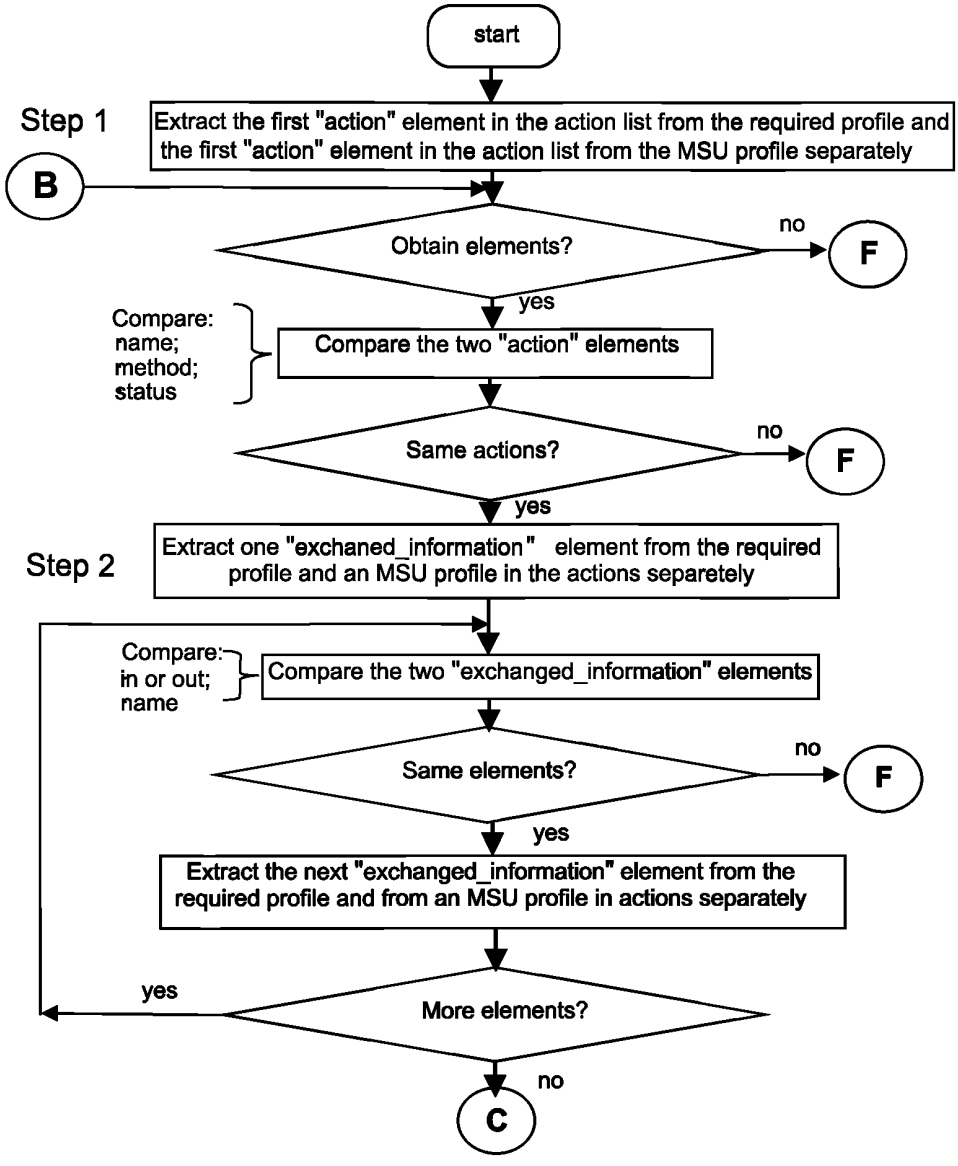
На рисунке G.1 имеются контуры двух видов: наружный контур и внутренний контур. В наружном контуре, контур B используется для сравнения операций в соответствии с установленным порядком требуемого профиля. Порядок установлен для объектов, упорядоченных по времени "**Time_Order_Of_MDD_object**", и объектов, расположенных в порядке поступления "**Event_Order_Of_MDD_object**". Элементы «производственной деятельности» упорядочены по времени для объектов **«Time_Order_Of_MDD_objects»** описания **«MDD_Description»**. Аналогично, элементы «производственной деятельности» упорядочены в порядке поступления для объектов «**Event_Order_Of_MDD_object**» описания **«MDD_Description»**. Сравнение элементов «производственной деятельности» выполяется строго в установленном порядке. Исключением являются объекты «**Set_Of_MDD_objects**», где каждый элемент производственной деятельности в требуемом профиле последовательно сравнивается со всеми элементами производственной деятельности в профиле MSU до достижения желаемого результата.

Внутренний контур фактически состоит из 4-х контуров типа B. Первый внутренний контур используется для сравнения с двумя элементами **MDD_name_action** из двух индивидуальных профилей. В соответствии с формальной структурой, показанной на рисунке 2, после сравнений двух элементов производственной деятельности рассматриваются соответствующие названия, методы и статусы.

Второй внутренний контур используется для сравнения элементов обмена информации **MDD_name_exchanged_information** для двух операций. В соответствии с формальной структурой, показанной на рисунке 2, каждый элемент обмена информации **exchanged_information** требуемого профиля последовательно сравнивается со всеми элементами обмена информации **exchanged_information** профиля MSU до получения желаемого результата. Так как может быть несколько элементов **exchanged_information** в одной операции, то после каждого такого сравнения рассматриваются входная информация **information_in,** выходная информация **information_out** и названия элементов.

Третий внутренний контур используется для сравнения элементов ограничений имени объекта данных **MDD_name_constraint** в двух операциях. В соответствии с формальной структурой, показанной на рисунке 2, каждый элемент ограничения в требуемом профиле последовательно сравнивается со всеми элементами ограничений в профиле MSU до получения желаемого результата. Так как может быть несколько ограничений в одной операции, то после каждого сравнения ограничений рассматривается название ограничения.

Четвертый внутренний контур используется для сравнения элементов ресурса объекта данных **MDD_name_resource** в двух операциях. В соответствии с формальной структурой, показанной на рисунке 2, каждый элемент ресурса в требуемом профиле последовательно сравнивается со всеми элементами ресурса в профиле MSU до получения желаемого результата. Так как может быть несколько ресурсов в одной операций, то после сравнения каждого ресурса рассматривается название этого ресурса.

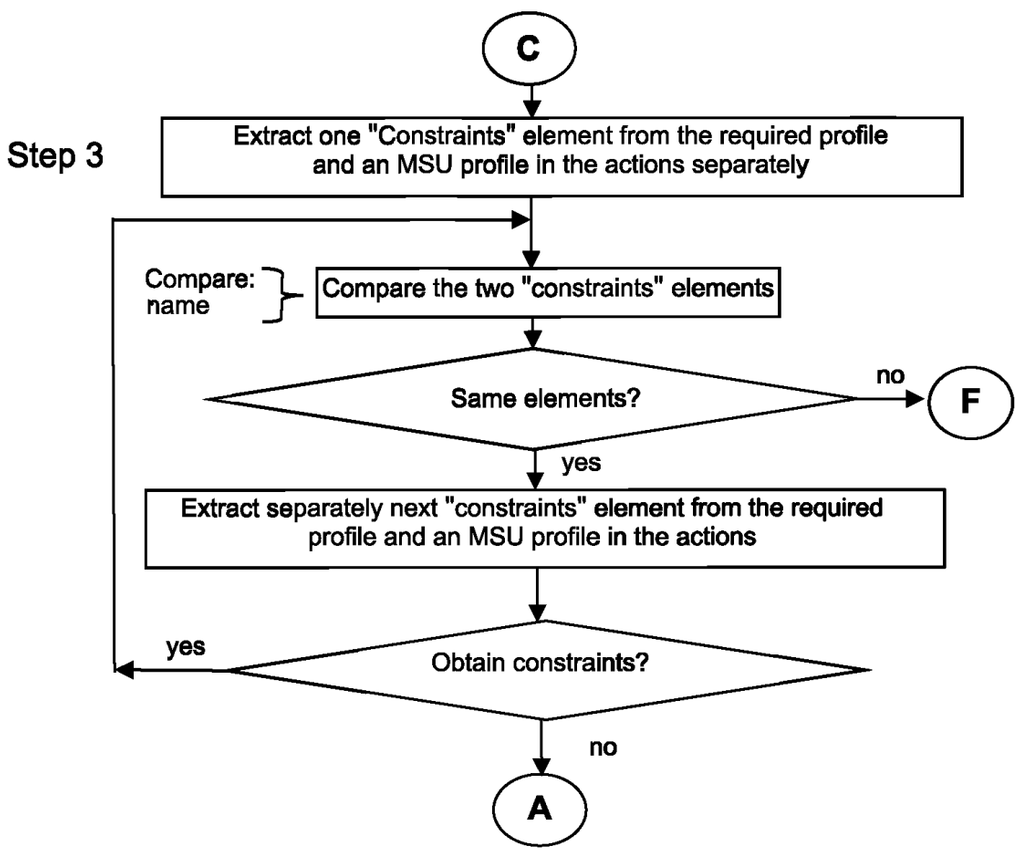


См. ниже продолжение.

start	Начало
Step 1	Шаг 1
Extract the first «action» element in the action list from the required profile and the first «action» element in the action list from the MSU profile separately	Извлечь в отдельности элемент первой операции списка операций из требуемого профиля и элемент первой операции списка операций из профиля MSU
Obtain elements?	Получены ли элементы?
no	Нет
yes	Да
Compare: - name; - method; - status	Сравнить: - имя - метод - статус
Compare the two «action» elements	Сравнение элементов двух операций
Same actions?	Одинаковы ли операции?
Step 2	Шаг 2

Extract one «exchanged information» element from the required profile and an MSU profile in the actions separately	Извлечь в отдельности в каждой операции один элемент «обмена информации» из требуемого профиля и один элемент «обмена информации» из профиля MSU
Compare: - in or out; - name	Сравнить: - вход или выход - имя
Compare the two «exchanged_information» elements	Сравнение указанных выше двух элементов «обмена информации»
Same elements?	Одинаковы ли эти элементы?
Extract the next «exchanged information» element from the required profile and from an MSU profile in the actions separately	Извлечь в отдельности в каждой операции следующий элемент «обмена информации» из требуемого профиля и один элемент «обмена информации» из профиля MSU
More elements?	Рассматривать ли другие элементы?

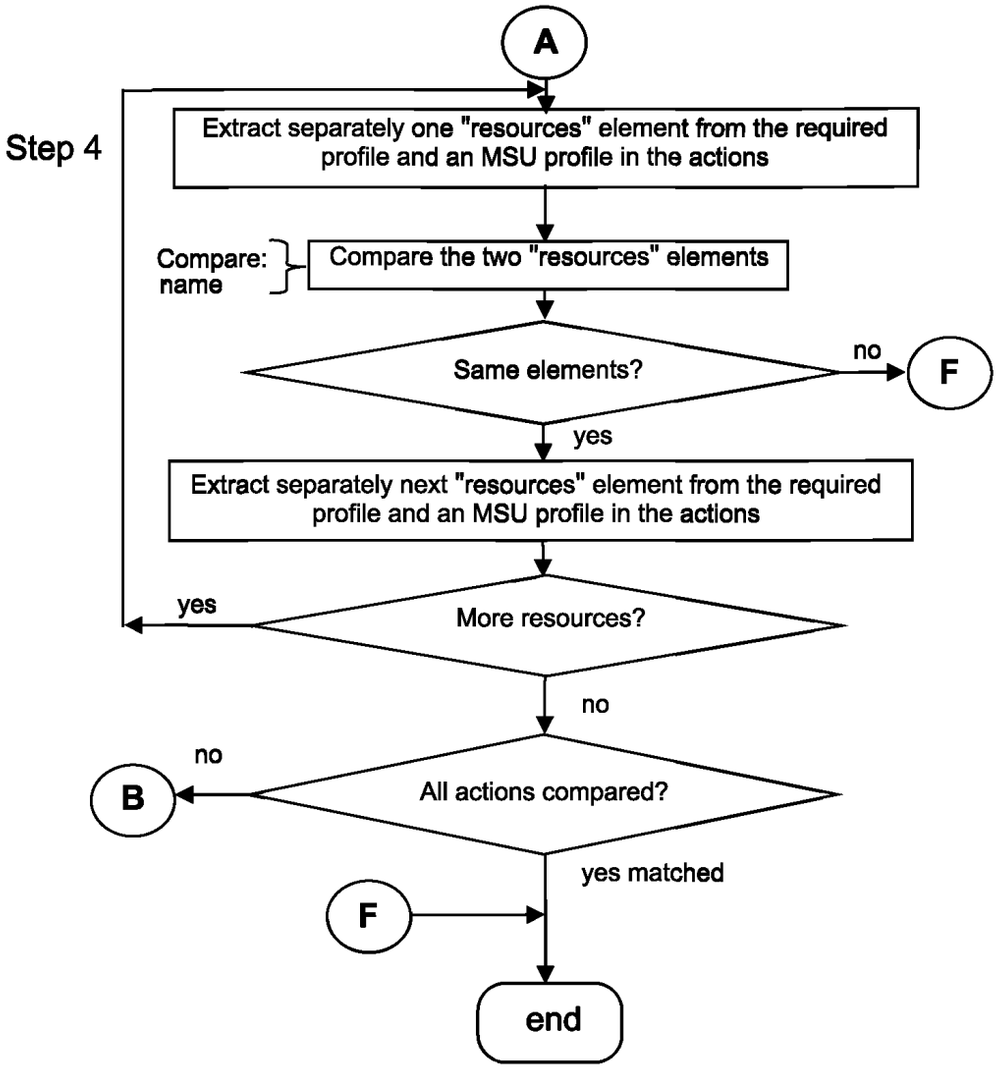
Примечание — A, B, C и F – точки соединения процедуры. Точка F указывает неудачное сопоставление.



См. ниже продолжение.

Step 3	Шаг 3
Extract one «Constraints» element from the required profile and an MSU profile in the actions separately	Извлечь в отдельности для каждой операции элемент «ограничения» из требуемого профиля и один соответствующий элемент из профиля MSU
Compare: - name	Сравнить: - имя
Compare the two «constraints» elements	Сравнение указанных двух элементов «ограничения»
Same elements?	Одинаковы ли эти элементы?
Extract separately next «constraints» element from the required profile and an MSU profile in the actions	Извлечь в отдельности для каждой операции следующие элементы «ограничения» из требуемого профиля и элементы из профиля MSU
Obtain constraints?	Получены ли ограничения?

Рисунок G.1, лист 2



Stap 4	Шаг 4
Extract separately one «resources» element from the required profile and an MSU profile in the actions	Извлечь в отдельности для каждой операции элемент «ресурсы» из требуемого профиля и элемент «ресурсы» из профиля MSU
Compare: - name	Сравнить: - имя
Compare the two «resources» elements	Сравнение двух указанных элементов «ресурсы»
Same elements?	Одинаковы ли элементы?
Extract separately next «resources» element from the required profile and an MSU profile in the actions	Извлечь в отдельности для каждой операции следующий элемент «ресурсы» из требуемого профиля и соответствующий элемент из профиля MSU
More resources?	Рассматривать ли ресурсы далее?
All actions compared?	Все ли операции прошли сравнение?
yes matched	Сопоставление закончено
End	Конец

Рисунок G.1 – Процедура сравнения описаний «comparison MDD_Description» для обнаружителей совпадений типа 2
Рисунок G.1, лист 3

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам Российской Федерации

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
ИСО 16100-1	IDT	ГОСТ Р ИСО 16100-1–2010 «Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 1. Структура»
ИСО 16100-2	IDT	ГОСТ Р ИСО 16100-2–2010 «Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 2. Методология профилирования»
ИСО 16100-3	IDT	ГОСТ Р ИСО 16100-3–2010 «Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 3. Службы интерфейса, протоколы и шаблоны возможностей»
ИСО 16100-4	IDT	ГОСТ Р ИСО 16100-4–2010 «Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 4. Методы аттестационных испытаний, критерии и отчеты»
ИСО 16100-5	IDT	ГОСТ Р ИСО 16100-5–2011 «Системы промышленной автоматизации и интеграция. Профилирование возможности интероперабельности промышленных программных средств. Часть 5. Методология согласования конфигураций профилей с помощью многоцелевых структур классов возможностей»
Примечание – В настоящей таблице использовано следующее условное обозначение степени соответствия стандартов: - IDT – идентичные стандарты.		

Библиография

[1] ИСО/МЭК 11179-5:2005 (ISO/IEC 11179-5:2005) Информационные технологии. Реестры метаданных (MDR). Часть 5. Принципы присвоения имен и идентификации (Information technology – Metadata registries (MDR) – Part 5: Naming and identification principles)

[2] ИСО 13584-24:2003 (ISO 13584-24:2003) Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 24. Логический ресурс. Логическая модель библиотеки поставщика (Industrial automation systems and integration – Parts library – Part 24: Logical resource: Logical model of supplier library)

[3] ИСО 13584-26:2000 (ISO 13584-26:2000) Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 26. Логический ресурс: идентификация поставщика информации (Industrial automation systems and integration - Parts library – Part 26: Logical resource: Information supplier identification)

[4] ИСО 13584-42:2010 (ISO 13584-42:2010) Системы промышленной автоматизации и интеграция. Библиотека данных на детали. Часть 42. Методология описания: методология структурирования групп деталей (Industrial automation systems and integration – Parts library – Part 42: Description methodology: Methodology for structuring parts families)

[5] ИСО 15745-1:2003 (ISO 15745-1:2003) Системы промышленной автоматизации и интеграция. Прикладная среда интегрирования открытых систем. Часть 1. Общее эталонное описание (Industrial automation systems and integration - Open systems application integration framework - Part 1: Generic reference description)

[6] ИСО/МЭК 19501:2005 (ISO/IEC 19501:2005) Информационные технологии. Открытая распределительная обработка. Унифицированный язык моделирования (UML). Версия 1.4.2 (Information technology - Open Distributed Processing – Unified Modeling Language (UML) Version 1.4.2)

[7] ИСО 22745-1:2010 (ISO 22745-1:2010) Промышленные автоматизированные системы и интеграция. Открытые технические словари и их применение к основным данным. Часть 1. Обзор и основные принципы (Industrial automation systems and integration –Open technical dictionaries and their application to master data – Part 1: Overview and fundamental principles)

[8] ИСО 22745-2:2010 (ISO 22745-2:2010) Промышленные автоматизированные системы и интеграция. Открытые технические словари и их применение к основным данным. Часть 2. Словарь (Industrial automation systems and integration –Open technical dictionaries and their application to master data – Part 2: Vocabulary)

[9] ИСО 22745-13:2010 (ISO 22745-13:2010) Промышленные автоматизированные системы и интеграция. Открытые технические словари и их применение к основным данным. Часть 13. Идентификация понятий и терминологии (Industrial automation systems and integration –Open technical dictionaries and their application to master data – Part 13: Identification of concepts and terminology)

[10] ИСО/ТС 22745-35:2010 (ISO/TS 22745-35:2010) Промышленные автоматизированные системы и интеграция. Открытые технические словари и их применение к основным данным. Часть 35. Запрос характеристических данных (Industrial automation systems and integration – Open technical dictionaries and their application to master data – Part 35: Query for characteristic data)

[11] ИСО/ТС 29002-5:2009 (ISO/TS 29002-5:2009) Промышленные автоматические системы и интеграция. Обмен характеристическими данными. Часть 5. Схема идентификации (Industrial automation systems and integration – Exchange of characteristic data – Part 5: Identification scheme)

[12] RFC 2276, Architectural Principles of Uniform Resource Name Resolution, IETF (Internet Engineering Task Force), ed. K. Sollins, 1998

УДК 658.52.011.56:006.354 ОКС 25.040.01 Т58

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

Редактор *В.М. Никифорова*
Технический редактор *А.Б. Заварзина*
Корректор *В.Г. Смолин*
Компьютерная верстка *Д.Е. Першин*

Сдано в набор 24.09.2015. Подписано в печать 8.10.2015. Формат 60x841/8. Гарнитура Ариал. Усл. печ. л. 8,84. Уч.-изд. л. 8,00. Тираж 30 экз. Зак. 3416.

Набрано в ООО «Академиздат».
gost.e-ecolog.ru gost.e-ecolog.ru

Издано и отпечатано во
ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru