ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

ПРЕДВАРИТЕЛЬНЫЙ НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ПНСТ 177—2016/ МЭК 62714-1—2014

ФОРМАТ ОБМЕНА ИНЖЕНЕРНЫМИ ДАННЫМИ ДЛЯ ИСПОЛЬЗОВАНИЯ В СИСТЕМАХ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ

Стандартизированный формат обмена данными AutomationML

Часть 1

Архитектура и общие требования

(IEC 62714-1:2014, IDT)

Издание официальное

Москва
Стандартинформ
2017

gost.e-ecolog.ru

Предисловие

1 ПОДГОТОВЛЕН ООО «НИИ экономики связи и информатики «Интерэкомс» (ООО «НИИ «Интерэкомс») на основе собственного перевода на русский язык англоязычной версии 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 6 декабря 2016 г. № 99-пнст

4 Настоящий стандарт идентичен международному стандарту IEC 62714-1:2014 «Формат обмена инженерными данными для использования в системах промышленной автоматизации. Стандартизированный формат обмена данными AutomationML. Часть 1. Архитектура и общие требования» (IEC 62714-1:2014 «Engineering data exchange format for use in industrial automation systems engineering — Automation markup language — Part 1: Architecture and general requirements», IDT).

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 и проведения его мониторинга установлены в ГОСТ Р 1.16-2011 (разделы 5 и 6).

Федеральное агентство по техническому регулированию и метрологии собирает сведения о практическом применении настоящего стандарта. Данные сведения, а также замечания и предложения по содержанию стандарта можно направить не позднее чем за девять месяцев до истечения срока его действия разработчику настоящего стандарта по адресу: 123423 Москва, ул. Народного Ополчения, д.32, и в Федеральное агентство по техническому регулированию и метрологии по адресу: 109074 Москва, Китайгородский проезд, д.7, стр. 1.

*В случае отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты» и журнале «Вестник Федерального агентства по техническому регулированию и метрологии». Уведомление будет размещено также на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Стандартинформ, 2017

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

Содержание

1 Область применения . . . 1
2 Нормативные ссылки . . . 1
3 Термины, определения и сокращения . . . 2
3.1 Термины и определения . . . 2
3.2 Сокращения . . . 3
4 Соответствие настоящему стандарту . . . 4
5 Спецификация архитектуры AutomationML . . . 4
5.1 Общие положения . . . 4
5.2 Архитектура языка AutomationML . . . 4
5.3 Версии документов языка AutomationML . . . 5
5.4 Метаинформация об инструментальных средствах источника AutomationML . . . 6
5.5 Идентификация объекта . . . 7
5.6 Спецификация отношений языка AutomationML . . . 9
5.7 Спецификация ссылки на документ языка AutomationML . . . 13
6 Базовые библиотеки языка AutomationML . . . 15
6.1 Введение . . . 15
6.2 Общие положения . . . 15
6.3 Библиотека класса интерфейсов AutomationML — AutomationMLInterfaceClassLib . . . 15
6.4 Базовая библиотека ролевых классов AutomationML — AutomationMLBaseRoleClassLib . . . 19
7 Моделирование пользовательских данных . . . 25
7.1 Общие положения . . . 25
7.2 Пользовательские атрибуты . . . 25
7.3 Пользовательский класс интерфейсов InterfaceClass . . . 26
7.4 Пользовательский ролевой класс RoleClass . . . 26
7.5 Пользовательский класс системных единиц SystemUnitClass . . . 28
7.6 Пользовательские иерархии экземпляров InstanceHierarchies . . . 28
8 Расширенные понятия языка AutomationML . . . 29
8.1 Общий положения . . . 29
8.2 Объект порт AutomationML Port . . . 29
8.3 Объект фасет AutomationML Facet . . . 30
8.4 Объект группа AutomationML Group . . . 30
8.5 Набор свойств AutomationML PropertySet . . . 31
8.6 Поддержка нескольких ролей . . . 31
8.7 Разбиение данных AutomationML верхнего уровня на различные документы . . . 34
8.8 Интернационализация . . . 34
8.9 Информация о версии объекта AutomationML . . . 34
Приложение А (справочное) Введение в язык AutomationML . . . 35
Приложение В (справочное) Представление библиотек AutomationML на языке XML . . . 64
Приложение ДА (справочное) Сведения о соответствии ссылочных международных стандартов национальным стандартам . . . 66
Библиография . . . 67

Введение

Комплекс стандартов МЭК 62714 определяет инженерное решение проблемы обмена данными для домена автоматизации производства.

Формат обмена данными, определенный в МЭК 62714 (язык разметки автоматизации AutomationML), — это формат обмена данными, основанный на XML языке. Он разработан для поддержки обмена данными на неоднородном множестве инструментальных средств инженерии.

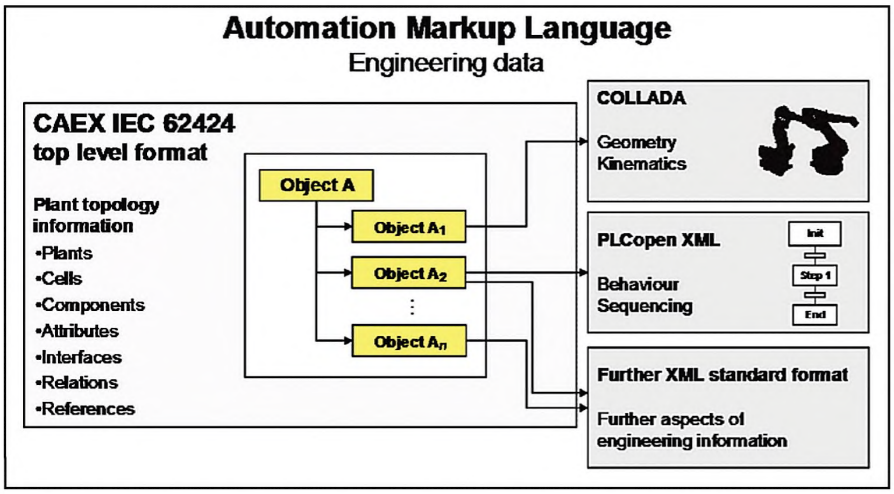
Цель языка AutomationML состоит в обеспечении взаимосвязи инструментальных средств инженерии в различных областях: проектирование механизированного оборудования, электротехническое проектирование, проектирование и управление производственными процессами, разработка человеко-машинного интерфейса (HMI), программирование логического контроллера (PLC), программирование роботов и т. д.

Язык AutomationML также предназначен для хранения инженерной информации в соответствии с объектно-ориентированной парадигмой, что позволяет моделировать физические и логические компоненты производственных объектов как объекты данных, инкапсулирующие различные аспекты. Указанные объекты могут состоять из вложенных объектов (подобъектов), которые сами могут являться частью композиции или агрегации. Типовые объекты автоматизированных установок включают в себя информацию о топологии, геометрии, кинематике и логике. При этом логика учитывает последовательность/упорядоченность, а также поведенческие и управленческие аспекты. Таким образом, целью обмена данными в инженерном домене является обмен объектно-ориентированными структурами данных топологии, геометрии, кинематики и логики.

Язык AutomationML комбинирует существующие промышленные форматы данных, используемые для хранения и обмена различными аспектами инженерной информации. Указанные форматы данных используются «как-есть» в рамках их собственных спецификаций вне зависимости от применения языка AutomationML.

Ядром языка AutomationML является формат данных верхнего уровня CAEX. Он соединяет различные форматы данных. Таким образом, язык AutomationML имеет унаследованную распределенную архитектуру документов.

Рисунок 1 иллюстрирует базовую архитектуру языка AutomationML, а также распределение информации о топологии, геометрии, кинематике и логике.



Automation Markup Language — язык разметки автоматизации AutomationML; Engineering data — инженерные данные; CAEX IEC 62424 top level format — формат верхнего уровня CAEX в соответствии с МЭК 62424; Plant topology information — информация о топологии производственной установки; Plants — производственные установки; Cells — производственные ячейки; Components — компоненты; Attributes — атрибуты; Interfaces — интерфейсы; Relations — соотношения; References — ссылки; Object A — объект A; Geometry — геометрия; Kinematics — кинематика; Behaviour — поведение; Sequencing — последовательность; PLCopen XML — XML язык для открытого программируемого логического контроллера; Further XML standard format — прочие стандартные форматы языка XML; Further aspects of engineering information — прочие аспекты инженерной информации

Рисунок 1 — Обзор формата обмена инженерными данными — языка AutomationML

Комплекс стандартов МЭК 62714 состоит из нескольких частей, распространяющихся на различные аспекты языка AutomationML:

- МЭК 62714-1: Архитектура и общие требования.

Данный стандарт устанавливает общую архитектуру языка AutomationML, порядок моделирования инженерных данных, классы, экземпляры, соотношения, ссылки, иерархии, базовые библиотеки AutomationML, расширенные понятия AutomationML. Данный стандарт является базой для всех последующих частей, устанавливает ссылочные механизмы на прочие второстепенные форматы.

- МЭК 62714-2: Библиотека ролевых классов.

Данный стандарт содержит описание дополнительных библиотек AutomationML.

- МЭК 62714-3: Геометрия и кинематика.

Данный стандарт устанавливает процедуру моделирования информации о геометрии и кинематике.

- МЭК 62714-4: Логика.

Данный стандарт устанавливает процедуру моделирования информации о логике, последовательности, поведении и управлении.

В будущем могут быть добавлены новые части, стандартизирующие возможности применения языка AutomationML для других форматов обмена данными.

В приложении А содержится справочная информация, рассматриваются примеры использования языка AutomationML.

В приложении В представлены библиотеки на XML языке, определенные в настоящем стандарте.

ПРЕДВАРИТЕЛЬНЫЙ НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФОРМАТ ОБМЕНА ИНЖЕНЕРНЫМИ ДАННЫМИ ДЛЯ ИСПОЛЬЗОВАНИЯ В СИСТЕМАХ ПРОМЫШЛЕННОЙ АВТОМАТИЗАЦИИ

Стандартизированный формат обмена данными AutomationML

Часть 1

Архитектура и общие требования

Engineering data exchange format for use in industrial automation systems engineering. Automation markup language. Part 1. Architecture and general requirements

Срок действия предстандарта — с 2017—06—01 до 2019—06—01

1 Область применения

Настоящий стандарт устанавливает общие требования и архитектуру языка AutomationML для моделирования инженерной информации, которой обмениваются между инструментами инженерии систем промышленной автоматизации и управления. Положения настоящего стандарта используются для экспорта/импорта приложений соответствующих инструментальных средств.

Настоящий стандарт не определяет требования к процедурам (практической реализации) обмена данными для соответствующих инструментальных средств импорта/экспорта.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие стандарты.

IEC 62424:2008, Representation of process control engineering — Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools (Представление технологии управления процессами. Запросы в диаграммах P&I и обмен данными между средствами P&I D и средствами PCE-CAE)

IEC 62714 (all parts), Engineering data exchange format for use in industrial automation systems engineering — Automation markup language (Формат обмена инженерными данными для использования в системах промышленной автоматизации. Стандартизированный формат обмена данными AutomationML (все части IEC 62714))

ISO/IEC 9834-8:2014, Information technology — Procedures for the operation of object identifier registration authorities — Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers (Информационные технологии. Процедуры для работы регистрационных органов идентификаторов объектов. Часть 8. Создание универсальных уникальных идентификаторов и их использование в идентификаторах объектов)

ISO/PAS 17506:2012, Industrial automation systems and integration — COLLADA digital asset schema specification for 3D visualization of industrial data (Системы промышленной автоматизации и интеграция. Спецификация цифровой схемы активов COLLADA для трехмерной визуализации промышленных данных)

COLLADA 1.4.1 :Март 2008, COLLADA — Схема цифрового актива. Выпуск 1.4.1 (см. gost.e-ecolog.ru khronos.org/файл/collada_spec_1_4.pdf).

Расширяемый язык разметки XML 1.0 1.0:2004, рекомендации W3C (см. gost.e-ecolog.ru REC-xml-20040204/).

Расширяемый язык разметки XML для открытого программируемого PLC контроллера open 2.0: Декабрь, №3 2008, а также PLCopen XML 2.0.1: Май №8 2009, Форматы расширяемого языка разметки XML для МЭК 61131-3 (см. gost.e-ecolog.ru /).

Издание официальное

3 Термины, определения и сокращения

3.1 Термины и определения

В настоящем стандарте используются следующие термины с соответствующими определениями:

3.1.1 **язык разметки автоматизации AutomationML** (AutomationML): Формат обмена данными на основе языка XML для инженерных данных производственного объекта в соответствии с МЭК 62714.

3.1.2 **объект автоматизации** (automation object): Физическая или логическая сущность в автоматизированной системе.

Примечание — Примером объекта автоматизации может быть компонент автоматизации (клапан, сигнал).

3.1.3 **объект (языка) AutomationML** (AutomationML object): Представление данных объекта автоматизации в соответствии с ролевыми классами языка AutomationML.

Примечание — Объекты языка AutomationML являются корневыми элементами языка AutomationML. Они представляют экземпляры, могут содержать управляющие элементы, атрибуты, интерфейсы, соотношения и ссылки.

3.1.4 **класс (языка) AutomationML** (AutomationML class): Предварительно определенный тип объекта языка AutomationML.

Примечание 1 — Классы языка AutomationML хранятся в библиотеках языка AutomationML.

Примечание 2 — Классы языка AutomationML определяют повторно применяемые инженерные решения, характеризуемые атрибутами, интерфейсами и агрегированными объектами.

Примечание 3 — Классы языка AutomationML могут быть использованы для множественного инстанцирования.

3.1.5 **атрибут (языка) AutomationML** (AutomationML attribute): Свойство, принадлежащее объекту языка AutomationML.

Примечание — Атрибуты языка AutomationML описаны в качестве элементов расширяемого языка разметки XML, соответствующие МЭК 62424:2008, раздел А.2.4.

3.1.6 **документ (языка) AutomationML** (AutomationML document): Документы в формате CAEX, соответствующие МЭК 62714 и включающие все ссылочные вспомогательные документы.

Примечание — Документы языка AutomationML могут храниться как файлы, строковые значения или потоки данных.

3.1.7 **файл (языка) AutomationML** (AutomationML file): Файлы в формате CAEX, соответствующие настоящему стандарту и имеющие расширение «.aml» за исключением ссылочных вспомогательных файлов.

3.1.8 **интерфейс (языка) AutomationML** (AutomationML interface): Отдельные точки соединения, принадлежащие объекту языка AutomationML и соединяемые с другим интерфейсом.

Примечание — Интерфейсы обеспечивают описание соотношений между объектами путем задания внутренних связующих элементов (внутренних ссылок) CAEX.

Пример — Интерфейс сигналов, интерфейс устройства, интерфейс питания.

3.1.9 **библиотека (языка) AutomationML** (AutomationML library): Библиотека, содержащая классы языка AutomationML.

3.1.10 **порт (языка) AutomationML** (AutomationML Port): Объект языка AutomationML, представляющий собой контейнер для группы интерфейсов, характеризуемых дополнительными свойствами.

Примечание — Порты принадлежат родительскому объекту языка AutomationML. Они описывают комплексные интерфейсы данного объекта. Порты могут соединяться друг с другом на более высоком уровне абстракции.

3.1.11 **группа (языка) AutomationML** (AutomationML Group): Объект языка AutomationML, определяющий общее для некоторой совокупности объектов языка AutomationML представление конкретных аспектов.

3.1.12 **фасет (языка) AutomationML** (AutomationML Facet): Объект языка AutomationML, определяющий общее для некоторой совокупности атрибутов (интерфейсов) языка AutomationML представление конкретных аспектов для одного объекта языка AutomationML.

3.1.13 **CAEX** (CAEX): Нейтральный формат данных, основанный на XML языке.

Примечание — CAEX — Это нейтральный формат данных, установленный в МЭК 62424:2008 (раздел 7, приложения A и C).

3.1.14 **соотношение «копии—экземпляры»** (copy-instance-relation): Соотношение между экземпляром и соответствующим классом, экземпляр которого создается путем копирования структур данных рассматриваемого класса.

Примечание — Экземпляр наследует все особенности и свойства класса—источника языка AutomationML. Модификация класса не ведет к модификации экземпляра. Внутри экземпляра свойства класса индивидуализируются. Последующее копирование становится возможным благодаря знанию класса—источника языка AutomationML.

3.1.15 **универсальный уникальный идентификатор** (universal unique identifier; UUID): Уникальный идентификатор объектов языка AutomationML.

3.1.16 **глобальный уникальный идентификатор** (global unique identifier; GUID): Практическая реализация UUID.

Примечание 1 — Конкретный пример идентификатора GUID: «{AC76BA86-7AD7-1033-7B44-A70000000000}».

Примечание 2 — В соответствии с МЭК 62714, идентификаторы GUID также представимы в краткой форме, например, «GUID1», «GUID2» и т. д. Это облегчает их читаемость и действует как конкретный GUID.

3.1.17 **соотношение наследования** (inheritance relation): Соотношение между двумя классами языка AutomationML.

Примечание — Производный класс наследует все атрибуты и особенности родительского класса.

3.1.18 **экземпляр** (instance): Представление данных индивидуального физического или логического элемента.

Примечание — Экземпляры могут быть расширены, например, с помощью агрегированных объектов или атрибутов.

3.1.19 **набор свойств PropertySet** (PropertySet): Класс стандартных ролей языка AutomationML, содержащий набор предварительно семантически определенных атрибутов.

3.1.20 **топология** (topology): Иерархическая структура системы, визуализируемая в виде древа объектов.

Примечание — Топология может включать несколько иерархий, пересеченных структур и сетей объектов.

3.1.21 **топология производственной установки** (plant topology): Иерархическая структура производственной установки, визуализируемая в виде древа объектов.

3.1.22 **публиковать** (publish): Моделирование структуры данных внешнего документа при использовании формата CAEX.

Примечание — Данное действие устанавливает определение соотношений между структурами данных независимых внешних документов.

3.1.23 **соотношение** (relation): Ассоциация между объектами формата CAEX.

Примечание — Примерами соотношений являются соотношения «потомок-родитель» и «класс-экземпляр класса».

3.1.24 **связующий элемент** (link): Обеспечивает связь между объектами типа CAEX ExternalInterface (внешний интерфейс).

Примечание — Связующий элемент моделируется с помощью сущностей CAEX InternalLinks (внутренние связи).

3.1.25 **ссылка** (reference): Ассоциация между сущностью CAEX InternalElements (внутренние элементы) и информацией, хранящейся во внешней среде.

3.2 Сокращения

В настоящем стандарте используются следующие сокращения:

AutomationML — язык разметки автоматизации (Automation Markup Language);

CAE — автоматизированное проектирование (Computer Aided Engineering);
CAEX — нейтральный формат обмена данными (Computer Aided Engineering eXchange);
COLLADA — совместное проектирование (Collaborative design activity);
GUID — глобальный уникальный идентификатор (Global unique identifier);
HMI — человеко-машинный интерфейс (Human machine interface);
ID — идентификатор (Identifier);
MES — автоматизированная система управления производством (Manufacturing execution system);
PLC — программируемый логический контроллер (Programmable logic controller);
URL — унифицированный указатель ресурса (Uniform resource locator);
URI — унифицированный идентификатор ресурса (Uniform resource identifier);
UUID — универсальный уникальный идентификатор (Universal unique identifier);
XML — расширяемый язык разметки (Extensible Markup Language).

4 Соответствие настоящему стандарту

Для обеспечения соответствия требованиям настоящего стандарта в части поддержания языка AutomationML необходимо выполнение требований, содержащихся в разделах 5, 6, 7 и 8 настоящего стандарта.

5 Спецификация архитектуры AutomationML

5.1 Общие положения

Ядром языка AutomationML является формат данных верхнего уровня CAEX. Это нейтральный формат данных, соответствующий требованиям МЭК 62424 (см. раздел 7, приложения A и C). Он соединяет установленные форматы инженерных данных по топологии, геометрии, кинематике, поведению систем и упорядоченности (последовательности) рассмотрения информации. Таким образом, базовой характеристикой языка AutomationML является унаследованная распределенная архитектура документов, фокусирующаяся на вышеуказанных инженерных аспектах.

Рисунки имеют исключительно иллюстративный характер. Графические представления не нормированы.

5.2 Архитектура языка AutomationML

В части общей архитектуры языка AutomationML, рассматриваются нижеследующие положения:

Информация о топологии производственной установки: топология производственной установки (далее — установки) применяется как структура данных верхнего уровня для инженерной информации об установке. Она моделируется с помощью формата данных CAEX в соответствии с МЭК 62424 (раздел 7, приложения A и C). Семантические расширения формата CAEX описаны отдельно. Множественные пересекающиеся иерархические структуры используются с помощью понятия зеркального объекта, определенного в МЭК 62424 (A.2.14). Зеркальные объекты не могут модифицироваться. Все изменения производятся на главном объекте (master object).

Примечание 1 — В соответствии с МЭК 62424 (раздел A.2.14), если один объект языка AutomationML называют «зеркальным объектом» по отношению ко второму объекту языка AutomationML, то указанный второй объект языка AutomationML называется «главным объектом». Зеркальный объект считается идентичным главному объекту. Это обеспечивает возможность разместить экземпляр одного объекта в различных иерархиях установки и, таким образом, позволяет моделировать сложные сети объектов с пересекающимися структурами.

Примечание 2 — МЭК 62714 не модифицирует синтаксически формат данных CAEX. В МЭК 62424 (раздел A.1.2 и приложение D) приведен справочный обзор и дополнительные примеры по топологии установки.

Информация о ссылках и соотношениях: ссылки и соотношения хранятся в соответствии с разделами 5.6 и 5.7. Соотношения между элементами внешней информации хранятся с помощью CAEX-механизмов. При необходимости, соответствующие партнеры по соединению/связи публикуются в описании топологии установки (в формате CAEX) с помощью внешних интерфейсов CAEX ExternalInterface. Они выводятся из классов стандартных интерфейсов языка AutomationML, указанных в разделе 6.3.

Примечание 3 — Ссылки отображают связи объектов CAEX с информацией, хранящейся во внешней среде. В разделе A.1.5 приведен справочный обзор рассматриваемых соотношений. Ссылки и публикации интерфейсов описаны в дополнительных частях МЭК 62714.

Примечание 4 — Соотношения отображают ассоциации между объектами CAEX.

Информация о геометрии и кинематике: требуемая информация о геометрии и кинематике хранится в формате данных COLLADA™[1]. Интерфейсы COLLADA, взаимосвязанные внутри формата верхнего уровня, должны быть опубликованы как внешние интерфейсы CAEX ExternalInterface.

Примечание 5 — МЭК 62714 не требует модификации синтаксиса формата данных COLLADA. Обзорный пример порядка выполнения ссылки COLLADA (см. раздел А.1.3). Подробности приведены в МЭК 62714-3.

Примечание 6 — С помощью информации о геометрии различных объектов COLLADA, полное представление формируется автоматически. Указанные файлы могут быть ссылочными из формата CAEX. Их взаимосвязь может быть обеспечена механизмами связи CAEX.

Информация о логике: информация о логике должна храниться в формате данных PLCopen XML. Если элементы логики (переменные, сигналы) должны быть взаимосвязаны внутри формата верхнего уровня, то они публикуются как внешние интерфейсы CAEX ExternalInterface. Все элементы языка PLCopen XML, опубликованные внутри формата верхнего уровня, должны иметь уникальный идентификатор внутри данного языка.

Примечание 7 — Информация о логике описывает последовательности действий и внутреннее поведение объектов, включающее входные/выходные связи и логические переменные. МЭК 62714 не требует модификации формата PLCopen XML. Обзор установленного порядка оформления ссылок на информацию о логике приведен в разделе А.1.4. Подробности см. МЭК 62714-4.

Ссылки на прочие форматы данных: МЭК 62714 может быть расширен в будущем путем публикации дополнительных частей, описывающих интеграцию последующих форматов данных XML, использующих механизмы ссылок языка AutomationML. Подробности приведены в дополнительных частях МЭК 62714.

Формат данных языка AutomationML не обеспечивает проверку совместимости ограничений, значений атрибутов, соотношений, ссылок или семантической корректности рассматриваемых данных. За это отвечает инструментальное средство (целевой инструмент), соответствующее приложение импорта/экспорта. Язык AutomationML содержит только синтаксическое подтверждение рассматриваемого документа в контексте с задействованными схемами.

5.3 Версии документов языка AutomationML

Каждый документ языка AutomationML, составленный на базе расширяемого языка разметки XML, должен хранить информацию о соответствующей настоящему стандарту версии языка AutomationML.

Примечание 1 — Нормативные положения в части информации о версии, относящейся к рассматриваемым экземплярам объектов языка AutomationML, определены в разделе 8.9. Порядок хранения специфической инструментальной метаинформации определен в разделе 5.4.

Нижеследующие положения определяют:

- корневой элемент CAEX «CAEXFile» для каждого документа верхнего уровня языка AutomationML должен иметь дочерний элемент CAEX «AdditionalInformation»;
- указанный элемент дополнительной информации «AdditionalInformation» должен иметь атрибут «AutomationMLVersion»;
- значение указанного атрибута «AutomationMLVersion» должно храниться в XML документе. В целях соответствия настоящему стандарту необходимо использовать версию «2.0»;
- каждый ссылочный документ CAEX должен соответствовать аналогичной версии языка AutomationML корневого документа. Смешивание документов с различными версиями AutomationML запрещено;
- каждый ссылочный внешний документ должен также соответствовать поименованным версиям схемы, определенным в вышеуказанной спецификации версии языка AutomationML. Смешивание внешних версий документа вне заданной спецификации версии AutomationML запрещено.

Рисунок 2 иллюстрирует текст расширяемого языка разметки XML, составленный для документа CAEX в соответствии с версией языка AutomationML 2.0.

---

[1] ™ COLLADA — это фирменный продукт группы Khronos. Данная информация дается для удобства пользователей настоящего стандарта. Она не является официальным согласованием указанного продукта стандартом МЭК. Эквивалентные продукты также могут быть использованы, если их применение дает аналогичные результаты.

```
<CAEXFile xmlns:xsi="http://gost.e-ecolog.ru                    lLSchema-instance" xsi:noNamespaceSchemaLocation="
CAEX_ClassModel.xsd" FileName="AutomationMLStandardLibrary2010-01-14_v1.99.aml" SchemaVersion="2.15">
  <AdditionalInformation AutomationMLVersion="2.0"/>
```

Рисунок 2 — Информация о версии документа AutomationML

- каждая стандартная библиотека AutomationML (каждая пользовательская библиотека AutomationML) должна содержать номер версии, использующий элемент CAEX «Version». Синтаксис значения номера версии в настоящем стандарте не определен;

- при необходимости, классы формата CAEX должны определять номер версии, использующий элемент CAEX «Version». Синтаксис и семантика номера версии классов внутри библиотеки AutomationML в настоящем стандарте не определены;

- одинаковые библиотеки различных версий не могут храниться в одном файле AutomationML;

Примечание 2 — Это гарантирует уникальность имени библиотеки AutomationML внутри файла AutomationML.

- разработчик документа AutomationML должен гарантировать, что ссылки производятся только на классы и внешние документы с совместимыми версиями.

МЭК 62714 основан на использовании нижеследующих форматов документов:

- CAEX версия 2.15;
- PLCopen XML версии 2.0 и 2.0.1;
- COLLADA версия 1.5.0 (в соответствии с ИСО/ПАС 17506) и COLLADA версия 1.4.1;
- стандартные библиотеки языка AutomationML в соответствии настоящим стандартом и последующими частями МЭК 62714.

5.4 Метаинформация об инструментальных средствах источника AutomationML

Для возможности передачи пользовательских данных из инструментального средства источника в целевое инструментальное средство необходимо хранить информацию об инструментальном средстве источника непосредственно в документе AutomationML. Нижеследующие положения определяют:

- каждый документ AutomationML должен содержать информацию об инструментальном средстве, использованном при оформлении документа AutomationML;

- в цепочке инструментальных средств обмена данных все принимающие участие инструментальные средства должны хранить данную информацию в документе CAEX аналогичным способом. Данный документ должен содержать информацию о нескольких инструментальных средствах рассматриваемой цепочки инструментальных средств обмена данными. Рассматриваемое инструментальное средство может удалить пользовательскую информацию о прочих инструментальных средствах. Это может затруднить итерационную процедуру обмена данными с прочими инструментальными средствами, в связи с чем, удаление пользовательской информации о прочих инструментальных средствах не рекомендуется;

- данная информация должна храниться как часть дополнительной информации CAEX AdditionalInformation корневого объекта документа CAEX;

- блок дополнительной информации AdditionalInformation должен быть поименован как заголовок «WriterHeader»;

- метаинформация должна содержать следующие данные:
  - имя экспортирующего программного обеспечения, пользовательского инструментального средства;
  - идентификатор ID пользовательского инструментального средства (он должен остаться неизменным);
  - информацию о продавце пользовательского инструментального средства;
  - URL размещении пользовательского инструментального средства;
  - информацию о номере версии пользовательского инструментального средства;
  - информацию о выпуске пользовательского инструментального средства;
  - информацию о времени внесения последнего изменения пользователем;
  - название проекта источника;
  - идентификатор проекта источника.

- содержание метаинформации определяется пользовательским инструментальным средством. Тип записи — xs:string;

- необходимая информация должна храниться с помощью атрибутов, указанных в таблице 2.

Таблица 2 — Метаинформация об инструменте источника AutomationML

Имя тэга XML	Тип	Уровень	Пример
WriterName	xs:string	Обязательный	«ToolX AML Exporter»
WriterID	xs:string	Обязательный	«ToolXToAML123»
WriterVendor	xs:string	Обязательный	«ToolX Vendor»
WriterVendorURL	xs:string	Обязательный	«gost.e-ecolog.ru
WriterVersion	xs:string	Обязательный	«0.2»
WriterRelease	xs:string	Обязательный	«123 prealpha»
LastWritingDateTime	xs:DateTime	Обязательный	«2011-05-25T09:30:47»
WriterProjectTitle	xs:string	«по выбору»	«eCarproduction»
WriterProjectID	xs:string	«по выбору»	«eCarproduction_LinePLC.prj»

Для представления метаинформации на расширяемом языке разметки XML используются нижеследующие положения:

- элемент «WriterHeader» должен быть дочкой XML элемента для элемента дополнительной информации CAEX AdditionalInformation корневого элемента CAEX;
- каждая единица метаинформации, поименованная в таблице 2, должна быть описана как дочка XML элемента «WriterHeader»;
- запрещено использовать несколько блоков метаинформации с одним именем в одном элементе «WriterHeader»;
- порядок представления метаинформации соответствует таблице 2.

Рисунок 3 приводит пример соответствующего описания на расширяемом языке разметки XML.

```
<AdditionalInformation>
  <WriterHeader>
    <WriterName>ToolX AML Exporter</WriterName>
    <ToolWriterID>ToolXToAML123</ToolWriterID>
    <WriterVendor>ToolX Vendor</WriterVendor>
    <WriterVendorURL>gost.e-ecolog.ru                ndor.org</WriterVendorURL>
    <WriterVersion>0.1</WriterVersion>
    <WriterRelease>123 prealpha</WriterRelease>
    <LastWritingDateTime>2013-12-31T12:00:00</LastWritingDateTime>
    <WriterProjectTitle>eCarproduction</WriterProjectTitle>
    <WriterProjectID>eCarproduction_LinePLC.prj</WriterProjectID>
  </WriterHeader>
</AdditionalInformation>
```

Рисунок 3 — Описание информации об инструментальном средстве источника AutomationML на языке XML

5.5 Идентификация объекта

Язык AutomationML соответствует объектно-ориентированной парадигме. Вся инженерная информация моделируется в виде объектов (принадлежит объектам). Вместе с тем, из-за неоднородности множества инструментами инженерии для идентификации объектов используются различные понятия. Например, уникальное имя, уникальный идентификатор, уникальный путь доступа. Одни средства позволяют изменять идентификатор по истечении срока существования, другие — нет. МЭК 62714 предоставляет возможность обмена данными между различными инструментами инженерии с учетом индивидуальной идентификации объекта. В соответствии с описанными характеристиками, настоящий стандарт компенсирует существующее разнообразие идентификационных понятий и определяет одно обязательное понятие идентификации объекта.

Для идентификации объекта используются нижеследующие положения:

- в соответствии с МЭК 62424 (раздел А.2.2.1) Классы языка AutomationML (ролевой класс RoleClass, класс интерфейсов InterfaceClass, класс системных единиц SystemUnitClass) идентифицируются тэгом «Name (имя)» формата CAEX;
- данное имя должно быть уникальным внутри рассматриваемого уровня иерархии соответствующей библиотеки AutomationML в течение всего срока существования класса;
- в соответствии с МЭК 62424 (раздел А.2.8) ссылка на классы производится с использованием полных путей доступа с помощью соответствующих сепараторов пути доступа;
- все экземпляры объекта AutomationML (внутренние элементы формата CAEX InternalElements, внешние интерфейсы формата CAEX ExternalInterface) идентифицируются соответствующими тэгами «ID» формата CAEX. Данные идентификаторы должны быть UUID идентификаторами в соответствии с ИСО/МЭК 9834-8.

Примечание 1 — Возможной практической реализацией UUID является глобальный уникальный идентификатор GUID.

Примечание 2 — В соответствии с МЭК 62424 (раздел А.3.15), тэг «ID» не является обязательным в отличие от настоящего стандарта.

Примечание 3 — В настоящем стандарте, идентификаторы UUID представляются также в краткой форме, например, «GUID1», «GUID2» и т. д.

Примечание 4 — Тэг «Name» формата CAEX указывает имя. Он имеет справочный характер и может изменяться с течением времени или при смене инструментального средства.

- если рассматриваемый идентификатор создан, то полученный UUID не может изменяться в течение всего срока существования соответствующего объекта для всех используемых инструментальных средств;
- ссылочные экземпляры используют значение «ID»;
- ссылка на интерфейс CAEX оформляется с использованием идентификатора UUID родительского объекта интерфейса, строки сепаратора «:» и имени экземпляра интерфейса;

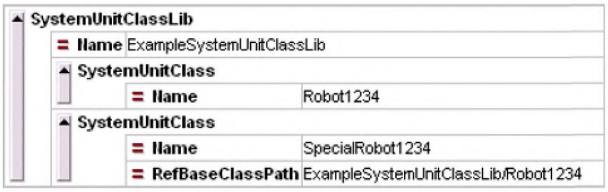
Пример 1 — «GUID:out».

- ссылка на атрибут CAEX оформляется с помощью соответствующего идентификатора UUID родительского объекта атрибута, строки сепаратора «.» и имени атрибута. Если рассматриваемый атрибут является вложенным, то за строкой сепаратора указывается подпуть доступа к атрибуту.

Пример 2 — «GUID.Colour».

Пример 3 — «GUID.Colour.red».

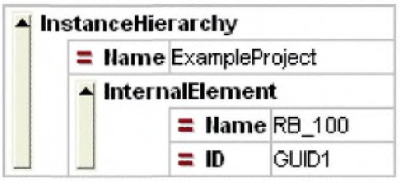
Рисунок 4 содержит пример библиотеки системных единиц SystemUnitClassLib, содержащей класс SystemUnitClass «Robot1234», а также класс SystemUnitClass «SpecialRobot1234», являющийся производным классом класса «Robot1234».




```
<SystemUnitClassLib Name="ExampleSystemUnitClassLib">
  <SystemUnitClass Name="Robot1234"/>
  <SystemUnitClass Name="SpecialRobot1234" RefBaseClassPath="ExampleSystemUnitClassLib/Robot1234"/>
</SystemUnitClassLib>
```

Рисунок 4 — Пример идентификации объекта класса AutomationML

Рисунок 5 содержит пример иерархии экземпляров InstanceHierarchy с одним объектом «RB_100», имеющим уникальный идентификатор, представленный строкой «GUID1».




```
<InstanceHierarchy Name="ExampleProject">
  <InternalElement Name="RB_100" ID="GUID1"/>
</InstanceHierarchy>
```

Рисунок 5 — Пример идентификации объекта экземпляра AutomationML

5.6 Спецификация отношений языка AutomationML

5.6.1 Общие положения

Рассмотрение указанных объектов необходимо для определения механизма обеспечения ассоциаций объектов друг с другом. В настоящем стандарте проводится различие между двумя механизмами хранения информации: ссылки и соотношения. Подраздел 5.6 акцентирует внимание на соотношениях, подраздел 5.7 на ссылках. Справочный обзор соотношений и ссылок приведен в разделе А.1.5.

5.6.2 Соотношения «родитель-потомок» между объектами языка AutomationML

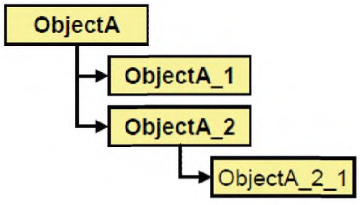
Соотношения типа «родитель-потомок» между экземплярами объектов используются для представления иерархических объектов структуры и описания структуры соотношений.

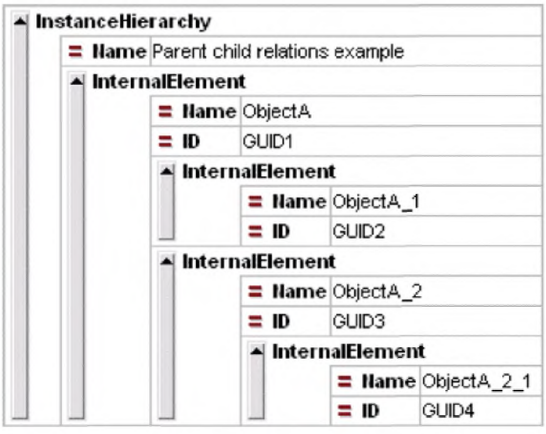
Для соотношений «родитель-потомок» между объектами AutomationML, используются нижеследующие положения:

- иерархия хранится в соответствии с МЭК 62424 (приложение А, раздел А.2.11).

Примечание — Кроме простых иерархий имеются и пересекающиеся иерархии (сети объектов). Они хранятся в соответствии с МЭК 62424 (раздел А.2.14).

Рисунок 6 содержит пример иерархии объектов и порядка ее хранения.






```
<InstanceHierarchy Name="Parent child relations example">
    <InternalElement Name="ObjectA" ID="GUID1">
        <InternalElement Name="ObjectA_1" ID="GUID2"/>
        <InternalElement Name="ObjectA_2" ID="GUID3">
            <InternalElement Name="ObjectA_2_1" ID="GUID4"/>
        </InternalElement>
    </InternalElement>
</InstanceHierarchy>
```

ObjectA — Объект А; InstanceHierarchy — Иерархия экземпляров; Name|Parent child relations example — Имя: пример соотношений «родитель—потомок»; Name|ObjectA — Имя: объект А; ID | GUID1 — Идентификатор: GUID1; InternalElement — Внутренний элемент

Рисунок 6 — Пример соотношения «родитель—потомок» между объектами AutomationML

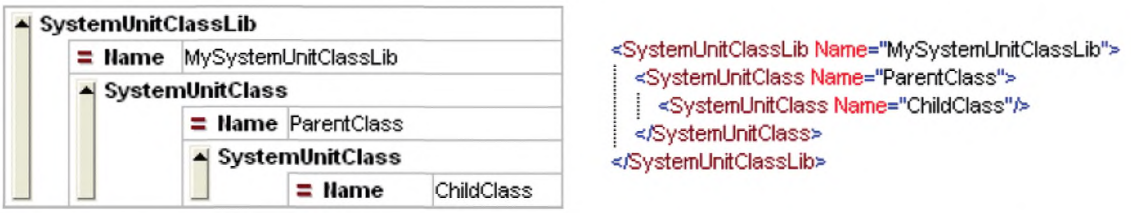
5.6.3 Соотношения «родитель—потомок» между классами языка AutomationML

Для соотношений «родитель—потомок» между классами AutomationML используются нижеследующие положения:

- соотношение «родитель—потомок» между классами AutomationML описывает только их иерархическое соседство. Это позволяет дать определения любым пользовательским иерархическим структурам;
- данное соотношение не имеет последующей семантики.

Примечание — Соотношение «родитель—потомок» не подразумевает наличия соотношения наследования. Соотношение наследования моделируется явно в соответствии с разделом 5.6.4.

Рисунок 7 содержит пример соотношения «родитель—потомок» между классами «ParentClass» и «ChildClass». «ChildClass» не обеспечивает соотношения наследования с родителем.



SystemUnitClassLib — Библиотека классов системных единиц; SystemUnitClass — Класс системных единиц

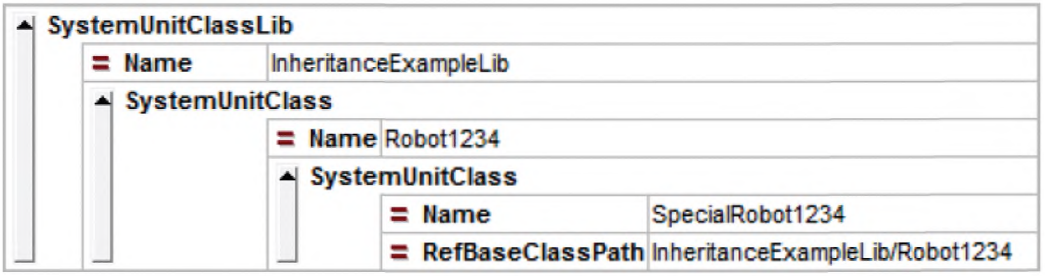
Рисунок 7 — Пример соотношения «родитель—потомок» между классами

5.6.4 Соотношение наследования

Для соотношений наследования используются нижеследующие положения:

- наследование между классами определяется в соответствии с МЭК 62424:2008 (раздел А.2.7);
- если наследование необходимо, то родительский класс указывается с помощью тэга CAEX «RefBaseClassPath», включающего полный путь доступа к базовому классу в соответствии с МЭК 62424 (раздел А.2.7);
- если рассматриваемый родительский класс расположен одним уровнем иерархии выше указанного дочернего класса, то родительский класс указывается путем сохранения имени данного родительского класса в тэге CAEX «RefBaseClassPath» без указания полного пути доступа.

Рисунок 8 содержит пример класса «Robot1234», а также класса «SpecialRobot1234», наследующего класс «Robot1234».




```
<SystemUnitClassLib Name="InheritanceExampleLib">
  <SystemUnitClass Name="Robot1234">
    <SystemUnitClass Name="SpecialRobot1234" RefBaseClassPath="
InheritanceExampleLib/Robot1234"/>
  </SystemUnitClass>
</SystemUnitClassLib>
```

SystemUnitClassLib — Библиотека классов системных единиц; Name | InheritanceExampleLib — Имя: библиотека примеров наследования; SystemUnitClass — Класс системных единиц; RefBaseClassPath — Путь доступа к ссылочному базовому классу

Рисунок 8 — Пример соотношения наследования между двумя классами

В добавление к данному примеру, тэг CAEX «RefBaseClassPath» может относиться либо к «Inheritance-ExampleLib/Robot1234», либо к «Robot1234», так как родительские классы расположены на один уровень иерархии выше уровня класса «SpecialRobot1234».

5.6.5 Соотношение «класс — экземпляр класса»

Экземпляры характеризуются уникальным идентификатором и набором параметров.

Для соотношений «класс—экземпляр класса» используются нижеследующие положения:

- объект языка AutomationML моделируется в качестве сущности InternalElements в формате CAEX как часть иерархии CAEX InstanceHierarchy или класса SystemUnitClass;
- объект языка AutomationML может быть одиночным объектом без связи с классом SystemUnitClass.

Примечание 1 — Объект языка AutomationML связан со стандартным ролевым классом AutomationML.

Примечание 2 — Экземпляры могут не иметь связи с базовой ролью AutomationMLBaseRole, они являются пользовательскими объектами и не являются объектами AutomationML;

- если объект языка AutomationML соотносится как «класс—экземпляр класса» с классом SystemUnitClass, то он создается как копия данного класса SystemUnitClass, включая внутреннюю архитектуру класса и всю унаследованную информацию.

Примечание 3 — Класс SystemUnitClass используется как шаблон. Изменения в классе SystemUnitClass не переносятся автоматически на соответствующие объекты автоматизации. Более того, объект автоматизации может переноситься без информации о классе. Он содержит внутри себя всю надлежащую информацию;

- допускается расширение или сокращение данных об экземпляре по сравнению с его классом источника.

Примечание 4 — Класс источника может быть подходящей стартовой точкой для модели экземпляра;

- скопированный класс источника должен быть указан в тэге CAEX «RefBaseSystemUnitPath» рассматриваемого экземпляра для последующего использования. Данный тэг должен указывать полный путь доступа и имя класса источника.

Примечание 5 — Если класс источника экземпляра изменяется, то это не приводит к изменению экземпляра. Обновление экземпляров — это возможный функциональный инструмент, который в настоящем стандарте не рассматривается;

- изменение класса источника должно вести к появлению новой версии класса с другим именем. Внутри нового класса полный путь доступа к старой версии класса должен храниться в тэге CAEX «OldVersion».

Примечание 6 — Данное положение поддерживает отслеживание изменений по всем существующим версиям класса;

- наследование между классом SystemUnitClass и экземпляром объекта не допускается.

Примечание 7 — Рассматриваемый экземпляр может быть только копией соответствующего класса. Наследование между классами и экземплярами может быть использовано при их расширении;

- соотношение между экземпляром и классом RoleClass указывается в соответствии с МЭК 62424 (раздел A.2.7), с помощью атрибута пути доступа «RefBaseRoleClassPath» для надлежащего требования к роли RoleRequirements. В отличие от МЭК 62424 (раздел A.2.7), наследование здесь не допускается. Все спецификации ролевых классов RoleClass копируются в соответствующие объекты AutomationML;
- соотношение между интерфейсом CAEX ExternalInterface и классом InterfaceClass указывается в соответствии с МЭК 62424 (раздел A.2.7). В отличие от МЭК 62424, наследование здесь не допускается. Все спецификации классов InterfaceClass копируются в соответствующие объекты AutomationML.

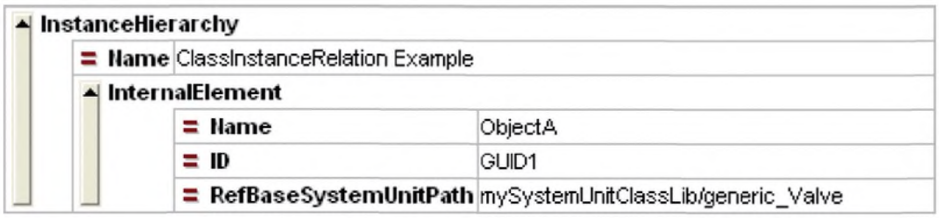
Рисунок 9 содержит пример соотношения «класс—экземпляр класса» между объектом «ObjectA» и пользовательским классом SystemUnitClass «generic_Valve».

В добавление к стандартным положениям соотношения «класс—экземпляр класса», нижеследующие специфические положения применяются в соответствии с зеркальным понятием CAEX:

- тэг «RefBaseSystemUnitPath» может указывать на другой экземпляр объекта вместо класса SystemUnitClass в соответствии с зеркальным понятием (см. МЭК 62424, A.2.14).

5.6.6 Соотношение «экземпляр-экземпляр»

Соотношение «экземпляр—экземпляр» — это соотношение между двумя интерфейсами произвольных объектов AutomationML.




```
<InstanceHierarchy Name="ClassInstanceRelation Example">
  <InternalElement Name="ObjectA" ID="GUID1" RefBaseSystemUnitPath="mySystemUnitClassLib/generic_Valve"/>
</InstanceHierarchy>
```

InstanceHierarchy — Иерархия экземпляров; InternalElement — Внутренний элемент; ID — Идентификатор; RefBaseSystemUnitPath — Ссылочный путь доступа к базовым системным единицам

Рисунок 9 — Пример соотношения «класс—экземпляр класса»

Для соотношений «экземпляр—экземпляр» используются нижеследующие положения:

- соотношения «экземпляр—экземпляр» хранятся в соответствии с МЭК 62424 (разделы A.2.5.3 и A.2.14) с помощью CAEX InternalLinks;

- CAEX InternalLinks хранится сущностью CAEX InternalElements, являющейся общим родителем самого нижнего уровня для соответствующих соединенных объектов CAEX;

- соотношения «экземпляр—экземпляр» определяются только между внешними интерфейсами CAEX ExternalInterface, принадлежащими соответствующим объектам AutomationML (см. МЭК 62424 (раздел A.2.3.1));

- сущность ExternalInterface выводится прямо или косвенно из одного из классов стандартных интерфейсов AutomationML;

Примечание 1 — Библиотека класса стандартных интерфейсов AutomationML рассмотрена в разделе 6.3. Класс интерфейсов определяет семантику интерфейса и, таким образом, семантику связующего элемента. Связующий элемент интерфейсов не имеет семантики, если нет ссылки на класс интерфейсов.

- документы COLLADA могут быть взаимосвязанными. Соответствующие интерфейсы COLLADA — это любые элементы, имеющие корректный идентификатор URI. Если рассматриваемые узлы должны быть взаимосвязанными в формате CAEX, то они должны быть опубликованы в CAEX путем добавления внешнего интерфейса CAEX ExternalInterface соответствующему объекту. Данный внешний интерфейс ExternalInterface выводится из класса стандартных интерфейсов AutomationML «COLLADAInterface» или из одного из его производных;

Примечание 2 — Стандартный класс интерфейсов «COLLADAInterface» рассмотрен в разделе 6.3.7. Подробности приведены в МЭК 62714-3.

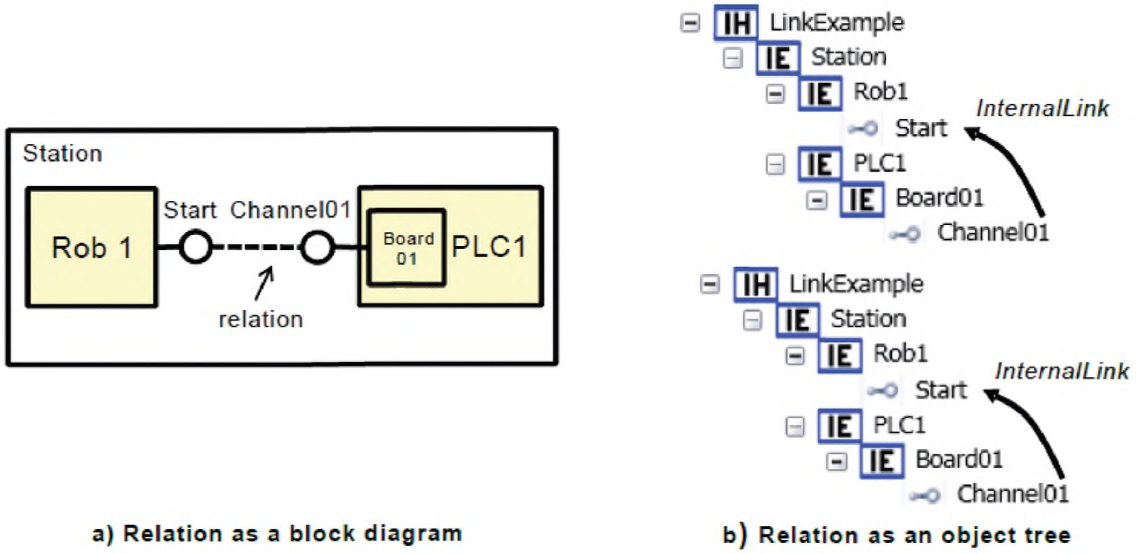
- документы формата PLCopen XML могут быть взаимосвязаны путем использования соответствующих интерфейсов PLCopen XML. Если в формате CAEX необходимо обеспечить взаимосвязь элементов PLCopen XML, то они должны быть опубликованы путем добавления сущности CAEX ExternalInterface в соответствующий объект. Данный внешний интерфейс ExternalInterface выводится из класса стандартных интерфейсов AutomationML «PLCopen-XMLInterface» или из одного из его производных.

Примечание 3 — Стандартный класс интерфейсов «PLCopenXMLInterface» рассмотрен в разделе 6.3.8. Подробности приведены в МЭК 62714-4.

Рисунок 10a) содержит пример, включающий робот «Rob1» и программируемый логический контроллер PLC «PLC1». Каждый имеет один интерфейс сигналов. Интерфейсы соединены. Рисунок 10b) отображает данный пример в виде иерархии объектов.

Рисунок 11 отображает представление данного примера на языке AutomationML. Ниже приведено описание на языке XML иерархии экземпляра InstanceHierarchy для данного примера, включающего все объекты AutomationML типа «Station», «Rob1», «PLC1» и «Board01», а также их интерфейсы.

Примечание 4 — Строки пути доступа для данного примера для улучшения их читаемости сокращены (обозначение «/.../»).



a) Relation as a block diagram **b) Relation as an object tree**

a) Relation as a block diagram — a) Соотношение в виде «блок—схема»; b) Relation as an object tree — b) Соотношение в виде «древа объектов»; Station — Станция; Start — Старт; Channel01 — Канал 01; Board 01 — Панель 01; relation — Соотношение; LinkExample — Пример связующего элемента; InternalLink — Внутренняя связь

Рисунок 10 — Пример соотношений видов «блок—схема» и «древо объектов»

5.7 Спецификация ссылки на документ языка AutomationML

5.7.1 Общие положения

Ссылка на документ обеспечивает связь между одним объектом AutomationML и одним внешним документом, содержащим, например, информацию о геометрии, кинематике, последовательности. Механизм ссылок использует стандартный интерфейс AutomationML «ExternalDataConnector» или один из его производных.

5.7.2 Ссылка на документ COLLADA

Ссылки на документы COLLADA производятся с помощью класса стандартных интерфейсов AutomationML «COLLADAInterface» или одного из его производных. Данный класс рассмотрен в разделе 6.3.7. Подробности приведены в МЭК 62714-3.

5.7.3 Ссылка на документ PLCopen XML

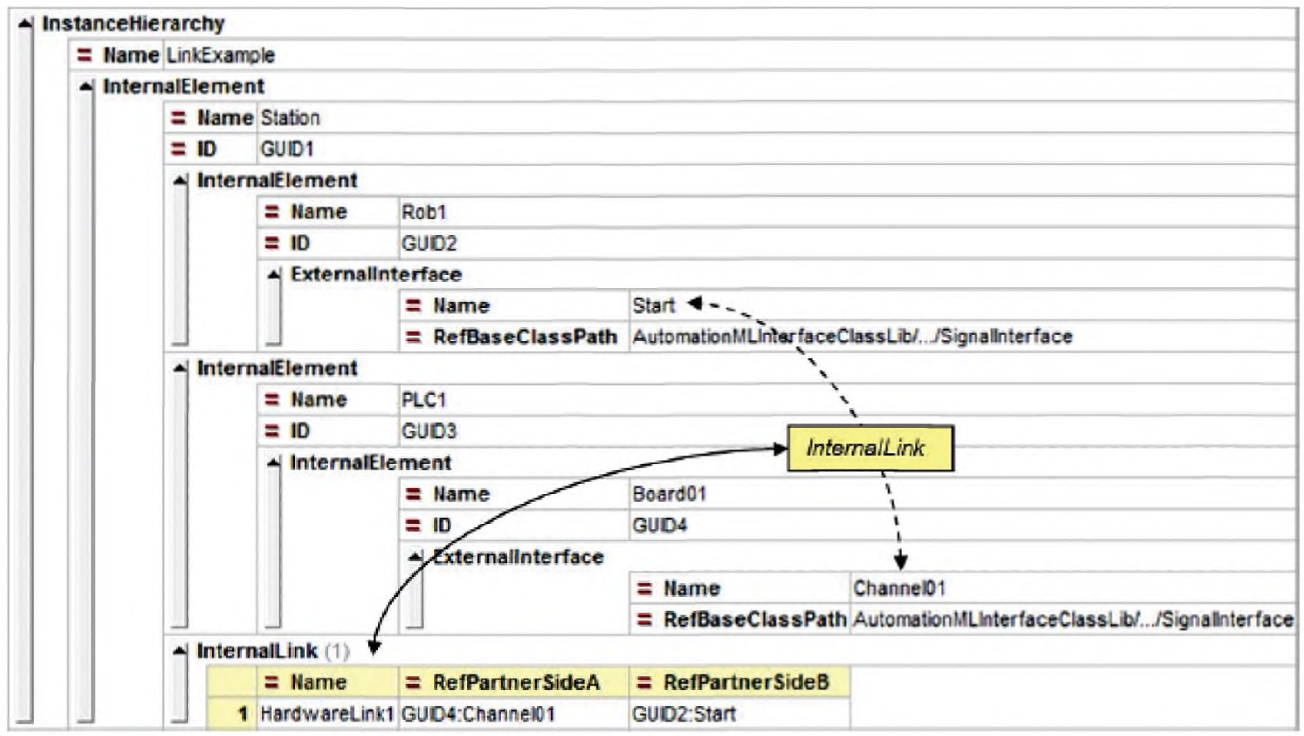
Ссылки на документы PLCopen XML производятся с помощью класса стандартных интерфейсов AutomationML «PLCopen-XMLInterface» или одного из его производных. Данный класс рассмотрен в разделе 6.3.8. Подробности приведены в МЭК 62714-4.

5.7.4 Ссылка на дополнительные документы

Будущие расширения МЭК 62714 могут установить дополнительные типы интерфейсов для ссылок на дополнительные типы документа. Они рассмотрены в отдельных частях МЭК 62714. В настоящем стандарте они не рассматриваются.

Для указанных расширений используются нижеследующие положения:

- если МЭК 62714 содержит дополнительные типы документа, то они моделируются с помощью дополнительных классов интерфейсов;
- указанные дополнительные интерфейсы моделируются как расширение библиотеки AutomationML InterfaceClass. Они выводятся непосредственно или косвенно из стандартного класса интерфейсов ExternalDataConnector;
- ссылки хранятся с помощью стандартных атрибутов стандартного класса интерфейсов;
- стандартный класс интерфейсов «ExternalDataConnector» используется для типов документа, включенных в МЭК 62714.




```
<InstanceHierarchy Name="LinkExample">
  <InternalElement Name="Station" ID="GUID1">
    <InternalElement Name="Rob1" ID="GUID2">
      <ExternalInterface Name="Start" RefBaseClassPath="AutomationMLInterfaceClassLib/.../ SignalInterface">
        <Attribute Name="Type">
          <Value>digital</Value>
        </Attribute>
        <Attribute Name="Direction">
          <Value>In</Value>
        </Attribute>
      </ExternalInterface>
    </InternalElement>
    <InternalElement Name="PLC1" ID="GUID3">
      <InternalElement Name="Board01" ID="GUID4">
        <ExternalInterface Name="Channel01" RefBaseClassPath="AutomationMLInterfaceClassLib/.../ SignalInterface">
          <Attribute Name="Type">
            <Value>digital</Value>
          </Attribute>
          <Attribute Name="Direction">
            <Value>Out</Value>
          </Attribute>
        </ExternalInterface>
      </InternalElement>
    </InternalElement>
    <InternalLink Name="HardwareLink1" RefPartnerSideA="GUID4:Channel01" RefPartnerSideB="GUID2:Start"/>
  </InternalElement>
</InstanceHierarchy>
```

```
<InstanceHierarchy Name="LinkExample">
  <InternalElement Name="Station" ID="GUID1">
    <InternalElement Name="Rob1" ID="GUID2">
      <ExternalInterface Name="Start" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
    </InternalElement>
    <InternalElement Name="PLC1" ID="GUID3">
      <InternalElement Name="Board01" ID="GUID4">
        <ExternalInterface Name="Channel01" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterfa
      </InternalElement>
    </InternalElement>
    <InternalLink Name="HardwareLink1" RefPartnerSideA="GUID4:Channel01" RefPartnerSideB="GUID2:Start"/>
  </InternalElement>
</InstanceHierarchy>
```

InstanceHirarchy — Иерархия экземпляра; Name | LinkExample — Имя: пример связующего элемента; InternalElement — Внутренний элемент; Name | Station — Имя: станция; ID | GUID1 — Идентификатор: GUID1; InternalElement — Внутренний элемент; Name — Имя; ID — Идентификатор; ExternalInterface — Внешний интерфейс; RefBaseClassPath — Ссылочный путь доступа базового класса; InternalLink — Внутренняя связь; ExternalInterface — Внешний интерфейс; HardwareLink1 — Аппаратная связь; RefPartnerSideA — Ссылочная сторона А партнера

Рисунок 11 — Пример соотношения между объектами «PLC1» и «Rob1»

6 Базовые библиотеки языка AutomationML

6.1 Введение

Раздел 6 определяет существенные базовые библиотеки AutomationML, а также базовые классы языка AutomationML, необходимые для моделирования корневых понятий AutomationML. Все описанные атрибуты являются частью стандартной библиотеки AutomationML. При необходимости они могут быть удалены из иерархии экземпляров InstanceHierarchy.

Примечание — Зависящие от домена библиотеки рассматриваются в последующих частях МЭК 62714.

6.2 Общие положения

Для базовых библиотек AutomationML, используются нижеследующие положения:

- все объекты AutomationML ассоциируются непосредственно или косвенно с ролевым классом «AutomationMLBaseRole»;
- все интерфейсы непосредственно или косвенно ассоциируются с классом интерфейсов AutomationML;
- при необходимости можно использовать атрибуты AutomationML, они могут быть удалены из объектов AutomationML.

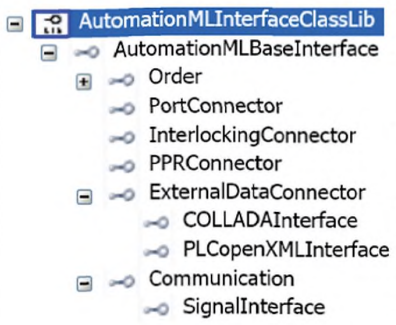
6.3 Библиотека класса интерфейсов AutomationML — AutomationMLInterfaceClassLib

6.3.1 Общие положения

Библиотека AutomationMLInterfaceClassLib моделируется в соответствии с МЭК 62424 (раздел 7, приложения A и C). МЭК 62714 использует понятие интерфейса CAEX. Допускаются пользовательские расширения данной библиотеки AutomationML в соответствии с разделом 7.3.

Каждый интерфейс выводится непосредственно или косвенно из класса нижеследующих стандартных библиотек AutomationMLInterfaceClassLib в соответствии с таблицей 3. Подразделы 6.3.2 — 6.3.10 описывают рассматриваемые классы интерфейсов в подробностях.

Таблица 3 — Библиотека класса интерфейсов AutomationMLInterfaceClassLib

Библиотека AutomationML InterfaceClass	Класс интерфейсов InterfaceClass	Описание
	AutomationMLBaseInterface	Абстрактный тип интерфейса
	Order	Интерфейс описания порядка
	PortConnector	Интерфейс описания портов
	PPRConnector	Коннектор для взаимосвязанных продуктов, ресурсов или процессов
	ExternalDataConnector	Характерный интерфейс коннектора внешних данных
	COLLADAInterface	Интерфейс документа COLLADA
	PLCopenXMLInterface	Интерфейс документа PLCopen XML
	Communication	Характерный коммуникационный интерфейс
	SignalInterface	Характерный интерфейс сигналов

На рисунке 12 приведена таблица, а на рисунке 13 описание стандартной библиотеки AutomationML Interface-ClassLib на языке XML. Подразделы 6.3.2 — 6.3.10 содержат подробную информацию о рассматриваемых классах.

- **InterfaceClassLib**
  - **Name**: AutomationMLInterfaceClassLib
  - **Description**: Standard AutomationML Interface Class Library
  - **Version**: 2.1.1
  - **InterfaceClass**
    - **Name**: AutomationMLBaseInterface
    - **InterfaceClass**
      - **Name**: Order
      - **RefBaseClassPath**: AutomationMLBaseInterface
      - **Attribute**
        - **Name**: Direction
        - **AttributeDataType**: xs:string
    - **InterfaceClass**
      - **Name**: PortConnector
      - **RefBaseClassPath**: AutomationMLBaseInterface
    - **InterfaceClass**
      - **Name**: InterlockingConnector
      - **RefBaseClassPath**: AutomationMLBaseInterface
    - **InterfaceClass**
      - **Name**: PPRConnector
      - **RefBaseClassPath**: AutomationMLBaseInterface
    - **InterfaceClass**
      - **Name**: ExternalDataConnector
      - **RefBaseClassPath**: AutomationMLBaseInterface
      - **Attribute**
        - **Name**: refURI
        - **AttributeDataType**: xs:anyURI
      - **InterfaceClass** (2)

	Name	RefBaseClassPath
1	COLLADAInterface	ExternalDataConnector
2	PLCopenXMLInterface	ExternalDataConnector

    - **InterfaceClass**
      - **Name**: Communication
      - **RefBaseClassPath**: AutomationMLBaseInterface
      - **InterfaceClass** (1)

	Name	RefBaseClassPath
1	SignalInterface	Communication

InterfaceClassLib — Библиотека класса интерфейсов; Name — Имя; Description Standard AutomationML Interface Class Library — Описание: Стандартная библиотека класса интерфейсов на языке AutomationML; Version — Версия; InterfaceClass — Класс интерфейсов; RefBaseClassPath — Ссылочный путь доступа к базовому классу; Attribute — Атрибут; Direction — Направление; AttributeDataType — Тип данных атрибута; Communication — Коммуникация; SignalInterface — Интерфейс сигналов

Рисунок 12 — Библиотека класса базовых интерфейсов AutomationML

```
<InterfaceClassLib Name="AutomationMLInterfaceClassLib">
  <Description>Standard AutomationML Interface Class Library</Description>
  <Version>2.1.1</Version>
  <InterfaceClass Name="AutomationMLBaseInterface">
    <InterfaceClass Name="Order" RefBaseClassPath="AutomationMLBaseInterface">
      <Attribute Name="Direction" AttributeDataType="xs:string"/>
    </InterfaceClass>
    <InterfaceClass Name="PortConnector" RefBaseClassPath="AutomationMLBaseInterface"/>
    <InterfaceClass Name="InterlockingConnector" RefBaseClassPath="AutomationMLBaseInterface"/>
    <InterfaceClass Name="PPRConnector" RefBaseClassPath="AutomationMLBaseInterface"/>
    <InterfaceClass Name="ExternalDataConnector" RefBaseClassPath="AutomationMLBaseInterface">
      <Attribute Name="refURI" AttributeDataType="xs:anyURI"/>
      <InterfaceClass Name="COLLADAInterface" RefBaseClassPath="ExternalDataConnector"/>
      <InterfaceClass Name="PLCopenXMLInterface" RefBaseClassPath="ExternalDataConnector"/>
    </InterfaceClass>
    <InterfaceClass Name="Communication" RefBaseClassPath="AutomationMLBaseInterface">
      <InterfaceClass Name="SignalInterface" RefBaseClassPath="Communication"/>
    </InterfaceClass>
  </InterfaceClass>
</InterfaceClassLib>
```

Рисунок 13 — Описание базовой библиотеки класса интерфейсов AutomationML на языке XML

6.3.2 Класс интерфейсов InterfaceClass AutomationMLBaseInterface

Таблица 4 содержит описание класса интерфейсов «AutomationMLBaseInterface».

Таблица 4 — Класс интерфейсов InterfaceClass AutomationMLBaseInterface

Имя класса	AutomationMLBaseInterface	
Описание	Класс интерфейсов «AutomationMLBaseInterface» — это базовый абстрактный тип интерфейса. Используется как родитель для описания всех классов интерфейсов AutomationML	
Родительский класс	Нет	
Атрибут	Нет	

6.3.3 Класс интерфейсов InterfaceClass Order

Таблица 5 содержит описание класса интерфейсов «Order « (порядок).

Таблица 5 — Класс интерфейсов InterfaceClass Order

Имя класса	Order	
Описание	Класс интерфейсов «Order» — это абстрактный класс, используемый для описания порядка следования элементов, например, последующий или предшествующий	
Родительский класс	AutomationMLInterfaceClassLib/AutomationMLBaseInterface	
Атрибут	Direction (тип= «xs:string»)	Атрибут «Direction» используется для описания направления. Допустимые значения: «In» (вход), «Out» (выход), «InOut» (вход/выход)

6.3.4 Класс интерфейсов InterfaceClass PortConnector

Таблица 6 содержит описание класса интерфейсов «PortConnector (коннектор порта)».

Таблица 6 — Класс интерфейсов InterfaceClass PortConnector

Имя класса	PortConnector	
Описание	Класс интерфейсов «PortConnector» используется для обеспечения соотношений высокого уровня между портами. Обзор понятия «порт» см. в разделе А.2.2	
Родительский класс	AutomationMLInterfaceClassLib/AutomationMLBaseInterface	
Атрибут	Нет	

6.3.5 Класс интерфейсов InterfaceClass PPRConnector

Таблица 7 содержит описание класса интерфейсов «PPRConnector».

Таблица 7 — Класс интерфейсов InterfaceClass PPRConnector

Имя класса	PPRConnector	
Описание	Класс интерфейсов «PPRConnector» обеспечивает соотношение между ресурсами, продуктами и процессами. См. дополнительную информацию в разделе А.2.6	
Родительский класс	AutomationMLInterfaceClassLib/AutomationMLBaseInterface	
Атрибут	Нет	

6.3.6 Класс интерфейсов InterfaceClass ExternalDataConnector

Таблица 8 содержит описание класса интерфейсов коннектора внешних данных «ExternalDataConnector».

Таблица 8 — Класс интерфейсов InterfaceClass ExternalDataConnector

Имя класса	ExternalDataConnector	
Описание	Класс интерфейсов «ExternalDataConnector» — это базовый абстрактный тип интерфейса. Используется для описания интерфейсов коннектора, ссылающегося на внешние документы. Классы «COLLADAInterface» и «PLCopenXMLInterface» выводятся из данного класса. Все существующие и будущие классы коннекторов должны выводиться непосредственно или косвенно из данного класса	
Родительский класс	AutomationMLInterfaceClassLib/AutomationMLBaseInterface	
Атрибут	refURI (тип= «xs:anyURI»)	Атрибут «refURI» используется для хранения пути доступа к ссылочному внешнему документу

6.3.7 Класс интерфейсов InterfaceClass COLLADAInterface

Таблица 9 содержит описание класса интерфейсов «COLLADAInterface». Подробности приведены в МЭК 62714-3.

Таблица 9 — Класс интерфейсов InterfaceClass COLLADAInterface

Имя класса	COLLADAInterface	
Описание	Класс интерфейсов «COLLADAInterface» используется для ссылки на внешние документы COLLADA и для публикации интерфейсов, определенных внутри внешнего документа COLLADA. Подробности приведены в МЭК 62714-3	
Родительский класс	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/ExternalDataConnector	
Атрибут	Нет	

6.3.8 Класс интерфейсов InterfaceClass PLCopenXMLInterface

Таблица 10 содержит описание класса интерфейсов «PLCopenXMLInterface». Подробности приведены в МЭК 62714-4.

Таблица 10 — Класс интерфейсов InterfaceClass PLCopenXMLInterface

Имя класса	PLCopenXMLInterface	
Описание	Класс интерфейсов «PLCopenXMLInterface» используется для ссылки на внешние документы PLCopen XML или для публикации сигналов (переменных), определенных внутри описания логики формата PLCopen XML. Подробности приведены в МЭК 62714-4	
Родительский класс	AutomationMLBaseInterface/ExternalDataConnector	
Атрибут	Нет	

6.3.9 Класс интерфейсов InterfaceClass Communication

Таблица 11 содержит описание класса интерфейсов «Communication».

Таблица 11 — Класс интерфейсов InterfaceClass Communication

Имя класса	Communication	
Описание	Класс интерфейсов «Communication» — это абстрактный тип интерфейса. Используется для описания интерфейсов, связанных с коммуникацией. Последующие классы, связанные с коммуникацией, выводятся из данного класса непосредственно или косвенно	
Родительский класс	AutomationMLInterfaceClassLib/AutomationMLBaseInterface	
Атрибут	Нет	

6.3.10 Класс интерфейсов InterfaceClass SignalInterface

Таблица 12 содержит описание класса интерфейсов сигналов «SignalInterface».

Таблица 12 — Класс интерфейсов InterfaceClass SignalInterface

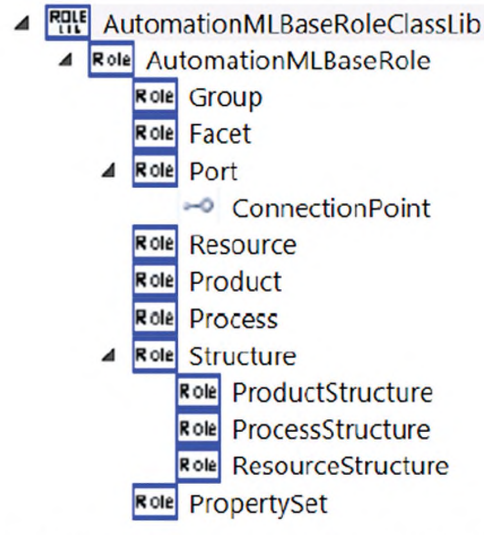
Имя класса	SignalInterface	
Описание	Класс интерфейсов «SignalInterface» используется для моделирования сигналов. Тип интерфейса — конфигурируемый. Он содержит описание цифровых и аналоговых входов и выходов, а также конфигурируемых входов/выходов. Пример приведен на рисунке 10	
Родительский класс	AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Communication	
Атрибут	Нет	

6.4 Базовая библиотека ролевых классов AutomationML— AutomationMLBaseRoleClassLib

6.4.1 Общие положения

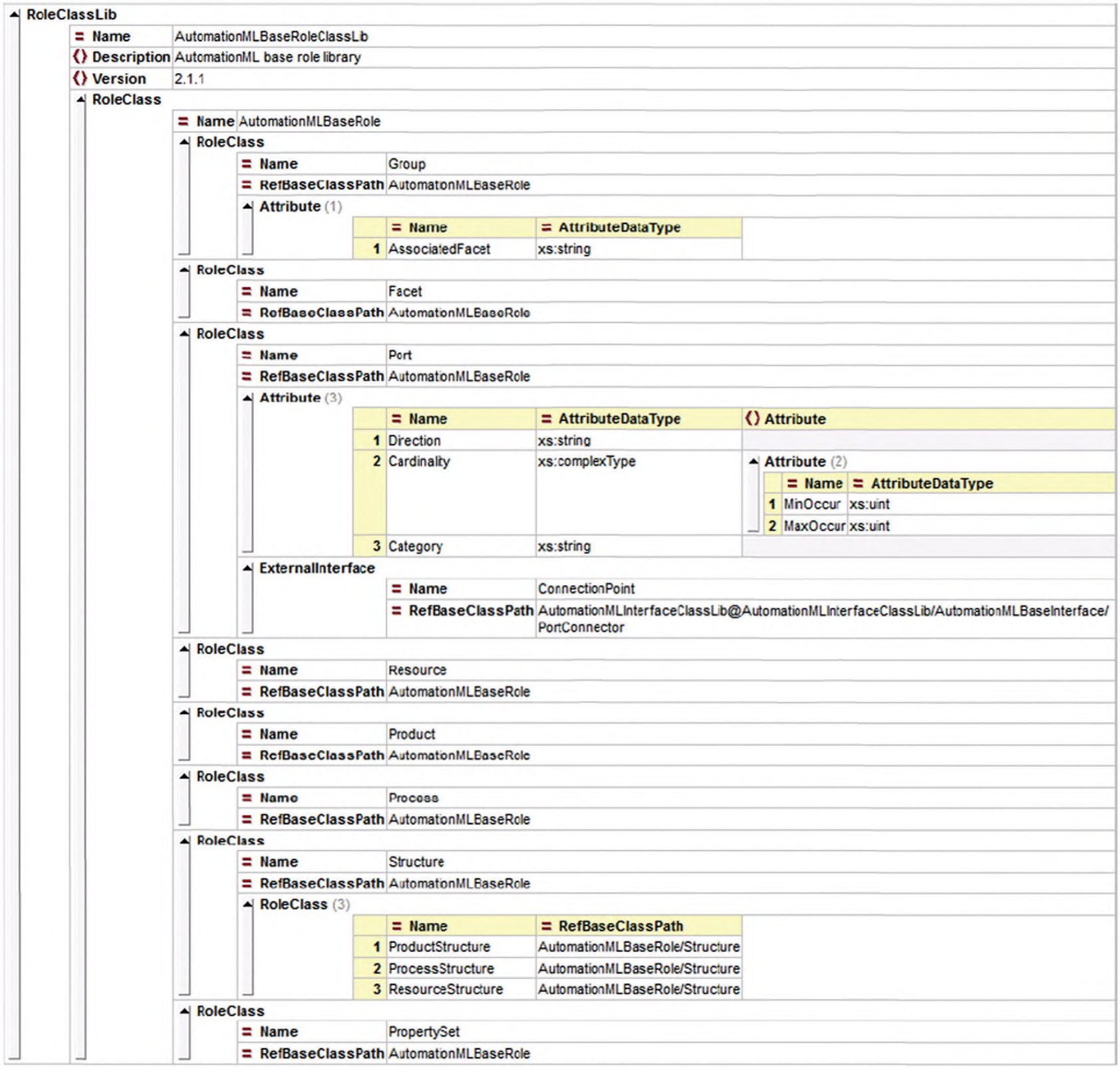
Подраздел 6.4 определяет базовую библиотеку AutomationML основных стандартных ролевых классов, необходимых для моделирования корневых понятий AutomationML. Роль — это класс, описывающий абстрактную функциональную возможность без определения основополагающей технической реализации. Примеры ролевых классов: «Resource», «Robot». Если ассоциировать ролевой класс с объектом AutomationML, то данный объект языка AutomationML приобретает семантику. Дополнительные расширенные библиотеки описаны в МЭК 62714-2. Все описанные атрибуты являются частью стандартной библиотеки AutomationML. Они могут быть удалены из иерархии экземпляров InstanceHierarchy, если в них нет необходимости.

Каждый объект языка AutomationML и каждый пользовательский ролевой класс должен прямо или косвенно ссылаться на одну из ролей указанной библиотеки AutomationML. Если рассматриваемая роль слишком специфична, то необходимо ссылаться на следующего родителя. На рисунках 14—16 представлен стандартный базовый ролевой класс RoleClass в виде древа объектов, XML таблицы, XML текста. Подробности по каждому ролевому классу приведены в разделах 6.4.2 — 6.4.13.



AutomationMLBaseRoleClassLib — Базовая библиотека ролевых классов; AutomationMLBaseRole — Базовая роль ; Group — Группа; Role Lib — Библиотека ролей; Role — Роль; Facet — Фасет; Port — Порт; ConnectionPoint — Точка соединения; Resource — Ресурс; Product — Продукт; Process — Процесс; Structure — Структура; ProductStructure — Структура продукта; ProcessStructure — Структура процесса; ResourceStructure — Структура ресурса; PropertySet — Набор свойств

Рисунок 14 — Базовая библиотека ролевых классов AutomationML



RoleClassLib — Библиотека ролевых классов; Name — Имя; Description | AutomationML base role library — Описание: Базовая библиотека ролей; Version — Версия; RoleClass — Ролевой класс; RefBaseClassPath — Ссылочный путь доступа к базовому классу; Attribute — Атрибут; AttributeDataType — Тип данных атрибута; Direction — Направление; Cardinality — Мощность множества; Category — Категория; ExternalInterface — Внешний интерфейс

Рисунок 15 — Базовая библиотека ролевых классов AutomationMLBaseRoleClassLib

6.4.2 Ролевой класс RoleClass AutomationMLBaseRole

Таблица 13 содержит описание класса базовых ролей «AutomationMLBaseRole».

Т а б л и ц а 13 — Ролевой класс RoleClass AutomationMLBaseRole

Имя класса	AutomationMLBaseRole	
Описание	Ролевой класс «AutomationMLBaseRole» — это базовый абстрактный тип ролей и базовый класс для всех стандартных или пользовательских ролевых классов	
Родительский класс	Нет	
Атрибут	Нет	

```
<RoleClassLib Name="AutomationMLBaseRoleClassLib">
  <Description>AutomationML base role library </Description>
  <Version>2.1.1</Version>
  <RoleClass Name="AutomationMLBaseRole">
    <RoleClass Name="Group" RefBaseClassPath="AutomationMLBaseRole">
      <Attribute Name="AssociatedFacet" AttributeDataType="xs:string"/>
    </RoleClass>
    <RoleClass Name="Facet" RefBaseClassPath="AutomationMLBaseRole"/>
    <RoleClass Name="Port" RefBaseClassPath="AutomationMLBaseRole">
      <Attribute Name="Direction" AttributeDataType="xs:string"/>
      <Attribute Name="Cardinality" AttributeDataType="xs:complexType">
        <Attribute Name="MinOccur" AttributeDataType="xs:uint"/>
        <Attribute Name="MaxOccur" AttributeDataType="xs:uint"/>
      </Attribute>
      <Attribute Name="Category" AttributeDataType="xs:string"/>
      <ExternalInterface Name="ConnectionPoint" RefBaseClassPath=
       "AutomationMLInterfaceClassLib@AutomationMLInterfaceClassLib/AutomationMLBaseInterface/PortConnector"/>
    </RoleClass>
    <RoleClass Name="Resource" RefBaseClassPath="AutomationMLBaseRole"/>
    <RoleClass Name="Product" RefBaseClassPath="AutomationMLBaseRole"/>
    <RoleClass Name="Process" RefBaseClassPath="AutomationMLBaseRole"/>
    <RoleClass Name="Structure" RefBaseClassPath="AutomationMLBaseRole">
      <RoleClass Name="ProductStructure" RefBaseClassPath="AutomationMLBaseRole/Structure"/>
      <RoleClass Name="ProcessStructure" RefBaseClassPath="AutomationMLBaseRole/Structure"/>
      <RoleClass Name="ResourceStructure" RefBaseClassPath="AutomationMLBaseRole/Structure"/>
    </RoleClass>
    <RoleClass Name="PropertySet" RefBaseClassPath="AutomationMLBaseRole"/>
  </RoleClass>
</RoleClassLib>
```

Рисунок 16 — Текст на языке XML для базовой библиотеки ролей AutomationMLBaseRoleClassLib

6.4.3 Ролевой класс RoleClass Group

Таблица 14 содержит описание ролевого класса «Group».

Таблица 14 — Ролевой класс RoleClass Group

Имя класса	Group	
Описание	Ролевой класс «Group» — это тип ролей для объектов, группирующих зеркальные объекты, которые соответствуют друг другу по определенным инженерным критериям. Объект группы AutomationML должен ссылаться на данную роль. Подробности и примеры рассмотрены в разделе 8.4	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Атрибут	«AssociatedFacet» (тип = «xs:string»)	Атрибут «AssociatedFacet» используется для определения имени соответствующего фасета. ***Пример — AssociatedFacet = «PLCFacet»***

6.4.4 Ролевой класс RoleClass Facet

Таблица 15 содержит описание ролевого класса «Facet».

Таблица 15 — Ролевой класс RoleClass Facet

Имя класса	Facet	
Описание	Ролевой класс «Facet» — это тип ролей для объектов, определяющих дополнительную точку зрения на атрибуты или интерфейсы объекта AutomationML. Объекты фасета AutomationML должны ссылаться на данную роль. Подробности и примеры приведены в разделе 8.3	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Атрибут	Нет	

6.4.5 Ролевой класс RoleClass Port

Таблица 16 содержит описание ролевого класса «Port».

Таблица 16 — Атрибуты «по выбору» для объектов ролевых классов AutomationML Port

Имя класса	Port	
Описание	Ролевой класс «Port» — это тип ролей для объектов, группирующих несколько интерфейсов и позволяющих описывать сложные интерфейсы в указанном смысле. Объект класса AutomationML Port должен ссылаться на данную роль. Подробности и примеры рассмотрены в разделе 8.2	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Атрибут	Direction (тип = «xs:string»)	Данный атрибут описывает направление класса Port. Значение атрибута выбирается из ряда: «In» (вход), «Out» (выход), «InOut» (вход/выход). Порты с направлением «In» могут соединяться только с портами с направлениями «Out» и «InOut». Порты с направлением «Out» могут соединяться только с портами с направлениями «In» и «InOut». Порты с направлением «InOut» могут соединяться с портами произвольного направления. Пример: Direction = «Out» (например, вилка) Direction = «In» (например, розетка) Direction = «InOut». Данная информация может быть использована, например, для валидации соединения. Примечание — Валидация рассматриваемого соединения лежит вне области применения МЭК 62714, но относится к функциональности инструментального средства
	«Cardinality»	Данный атрибут является комплексным и не должен иметь значения. Соответствующие второстепенные атрибуты описаны в таблице 17
	«Category» (тип = «xs:string»)	Данный атрибут описывает тип порта. Значение данного атрибута определяется пользователем. Соединять можно только порты с одинаковым значением атрибута Category. ***Пример — Category = «МатериалFlow».***

Атрибут «Cardinality» (мощность множества) имеет два второстепенных атрибута (субатрибуты), описанных в таблице 17.

Таблица 17 — Второстепенные атрибуты для атрибута «Cardinality»

Атрибут	Тип	Описание	Пример
«MinOccur»	xs:unsignedInt	Значение атрибута MinOccur указывает минимально возможное количество соединений, входящих в данный порт (исходящих из данного порта). Значение атрибута должно быть «больше чем или равно» 0	MinOccur = 1 Это означает, что данный порт должен соединяться, по крайней мере, с одним портом
«MaxOccur»	xs:unsignedInt	Значение атрибута MaxOccur указывает максимально возможное количество соединений, входящих в данный порт (исходящих из данного порта). Значение данного атрибута должно быть «больше чем или равно» значению атрибута MinOccur или равно 0, что означает неопределенность	MaxOccur = 3 Это означает, что данный порт может быть соединен максимально с тремя другими портами

Дополнительно данный объект порта AutomationML должен иметь внешний интерфейс CAEX ExternalInterface, полученный из класса интерфейсов AutomationML InterfaceClass «PortConnector» (см. таблицу 18).

Примечание — Данный интерфейс позволяет соединять рассматриваемый порт с некоторым количеством других портов на абстрактном уровне без подробных описаний внутренних соотношений между субинтерфейсами (см. рисунок А.13).

Таблица 18 — Интерфейс класса портов AutomationML Port

Интерфейс	Тип	Описание	Пример
Имя определяется пользователем, например, «ConnectionPoint»	PortConnector	Данный CAEX интерфейс позволяет соединять класс Port с некоторым количеством других портов на абстрактном уровне. Внутренние соотношения между отдельными интерфейсами порта не рассматриваются	См. раздел А.2.2.2

6.4.6 Ресурс ролевых классов RoleClass Resource

Таблица 19 содержит описание ролевого класса «Resource».

Таблица 19 — Ролевой класс RoleClass Resource

Имя класса	Resource	
Описание	Ролевой класс «Resource» — это базовый абстрактный тип ролей и базовый класс для всех ролей ресурсов AutomationML. Он описывает установки, оборудование и другие производственные ресурсы. Объекты ресурса AutomationML должны непосредственно или косвенно ссылаться на данную роль. Пример рассмотрен в разделе А.2.6	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Атрибут	Нет	

Дополнительно, при необходимости, объекты ресурса AutomationML должны иметь внешний интерфейс CAEX ExternalInterface «PPRConnector» для создания соотношений с продуктами и процессами (см. раздел 6.3.5).

6.4.7 Ролевой класс RoleClass Product

Таблица 20 содержит описание ролевого класса «Product».

Таблица 20 — Ролевой класс RoleClass Product

Имя класса	Product	
Описание	Ролевой класс «Product» — это базовый абстрактный тип ролей и базовый класс для всех ролей продуктов AutomationML. Он описывает продукты, части продуктов и продукты, относящиеся к материалам, обработанным на описанной установке. Объекты продукта AutomationML должны непосредственно или косвенно ссылаться на данную роль. Пример рассмотрен в разделе А.2.6	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Атрибут	Нет	

Дополнительно, при необходимости, объекты продукта AutomationML должны иметь внешний интерфейс CAEX ExternalInterface «PPRConnector» для создания соотношений с ресурсами и процессами (см. раздел 6.3.5).

6.4.8 Ролевой класс RoleClass Process

Таблица 21 содержит описание ролевого класса «Process».

Таблица 21 — Ролевой класс RoleClass Process

Имя класса	Process	
Описание	Ролевой класс «Process» — это базовый абстрактный тип ролей и базовый класс для всех ролей процессов AutomationML. Он описывает производственные процессы. Объекты процессов AutomationML должны непосредственно или косвенно ссылаться на данную роль. Пример рассмотрен в разделе А.2.6	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Атрибут	Нет	

Дополнительно, при необходимости, объекты процессов AutomationML должны иметь внешний интерфейс CAEX ExternalInterface «PPRConnector» для создания соотношений с продуктами и ресурсами (см. раздел 6.3.5).

6.4.9 Ролевой класс RoleClass Structure

Таблица 22 содержит описание ролевого класса «Structure».

Таблица 22 — Ролевой класс RoleClass Structure

Имя класса	Structure	
Описание	Ролевой класс «Structure» — это базовый абстрактный тип ролей для объектов, являющихся структурными элементами в иерархии установки. Например, фальцевальная машина, производственная площадка, производственная линия. Объекты структуры AutomationML должны непосредственно или косвенно ссылаться на данную роль	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Атрибут	Нет	

6.4.10 Ролевой класс RoleClass ProductStructure

Таблица 23 содержит описание ролевого класса «ProductStructure».

Таблица 23 — Ролевой класс RoleClass ProductStructure

Имя класса	ProductStructure	
Описание	Ролевой класс «ProductStructure» — это абстрактный тип ролей для продукт-ориентированной иерархии объектов. Объект структуры продукта AutomationML должен непосредственно или косвенно ссылаться на данную роль	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure	
Атрибут	Нет	

6.4.11 Ролевой класс RoleClass ProcessStructure

Таблица 24 содержит описание ролевого класса «ProcessStructure».

Таблица 24 — Ролевой класс RoleClass ProcessStructure

Имя класса	ProcessStructure	
Описание	Ролевой класс «ProcessStructure» — это абстрактный тип ролей для процесс-ориентированной иерархии объектов. Объект структуры процесса AutomationML должен непосредственно или косвенно ссылаться на данную роль	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure	
Атрибут	Нет	

6.4.12 Ролевой класс RoleClass ResourceStructure

Таблица 25 содержит описание ролевого класса «ResourceStructure».

Таблица 25 — Ролевой класс RoleClass ResourceStructure

Имя класса	ResourceStructure	
Описание	Ролевой класс «ResourceStructure» — это абстрактный тип ролей для ресурс-ориентированной иерархии объектов. Объект структуры ресурса AutomationML должен непосредственно или косвенно ссылаться на данную роль.	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Structure	
Атрибут	Нет	

6.4.13 Ролевой класс RoleClass PropertySet

Таблица 26 содержит описание ролевого класса «PropertySet».

Таблица 26 — Ролевой класс RoleClass PropertySet

Имя класса	PropertySet	
Описание	Ролевой класс «PropertySet» — это абстрактный тип ролей, обеспечивающий определение наборов свойств, соответствующих определенному инженерному аспекту. Объект набора свойств AutomationML должен непосредственно или косвенно ссылаться на данную роль. Нормативные положения описаны в разделе 8.5. Подробности и примеры приведены в разделе А.2.5	
Родительский класс	AutomationMLBaseRoleClassLib/AutomationMLBaseRole	
Атрибут	Нет	

7 Моделирование пользовательских данных

7.1 Общие положения

Раздел 7 устанавливает порядок моделирования пользовательских данных в формате AutomationML. Моделирование специфических пользовательских данных — это основное понятие языка AutomationML. Пользовательские данные — это те атрибуты CAEX, классы интерфейсов InterfaceClass и классы ролей RoleClass, которые не были предварительно определены МЭК 62714. Механизм моделирования пользовательских данных обусловлен форматом CAEX для данных верхнего уровня языка AutomationML.

Для обмена пользовательскими данными, могут потребоваться специальные пользовательские соглашения и функциональные возможности, которые не рассматриваются в МЭК 62714. Специальная метаинформация об инженерных инструментальных средствах источника, описанная в разделе 5.4, поддерживает указанные функциональные возможности.

Язык AutomationML обеспечивает возможность определять соотношения между пользовательскими данными и стандартными данными с помощью понятия роли, понятия набора свойств PropertySet и стандартных отображений CAEX. Указанные понятия облегчают автоматическую интерпретацию пользовательских классов и атрибутов.

7.2 Пользовательские атрибуты

Все атрибуты, определенные в МЭК 62714, называются атрибутами AutomationML. Все атрибуты, не определенные в МЭК 62714, называются пользовательскими атрибутами. Атрибуты AutomationML и пользовательские атрибуты хранятся одинаковым образом в качестве CAEX атрибутов.

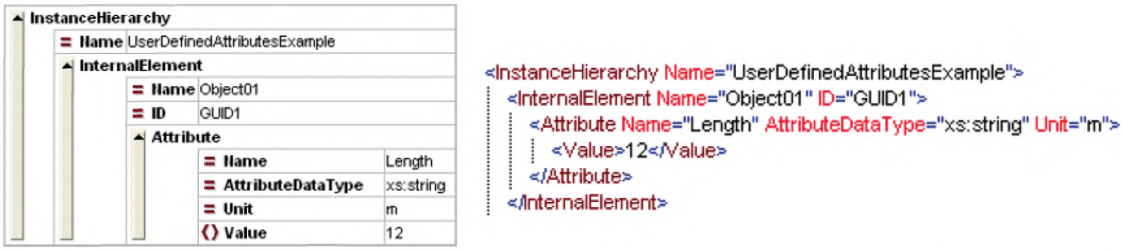
Для пользовательских атрибутов, используются нижеследующие положения:

- атрибуты CAEX должны храниться на языке AutomationML в соответствии с определением атрибутов CAEX, приведенным в МЭК 62424 (раздел А.2.4);

- если необходимы единицы измерения, то пользовательские атрибуты должны базироваться на уже имеющейся системе единиц. Системы единиц измерений в настоящем стандарте не рассматриваются.

Предлагается использовать систему единиц СИ в соответствии с ИСО 80000-1. Для единиц информационных технологий предлагается использовать МЭК 60027.

Рисунок 17 содержит пример пользовательского объекта «Object01» с пользовательским атрибутом «Length» (длина).



InstanceHierarchy — Иерархия экземпляров; Name — Имя; InternalElement — Внутренний элемент; ID — Идентификатор; Attribute — Атрибут; AttributeDataType — Тип данных атрибута; Unit — Единица измерения; Value — Значение

Рисунок 17 — Пример пользовательского атрибута

7.3 Пользовательский класс интерфейсов InterfaceClass

Все классы интерфейсов InterfaceClass, определенные в МЭК 62714, называются классами интерфейсов AutomationML InterfaceClass. Все классы интерфейсов, не определенные в МЭК 62714, называются пользовательскими классами интерфейсов InterfaceClass.

Для пользовательских классов интерфейсов InterfaceClass, используются нижеследующие положения:

- все пользовательские классы интерфейсов InterfaceClass должны храниться в соответствии с определением CAEX InterfaceClass, приведенным в МЭК 62424 (раздел A.2.5);

Примечание — Классы интерфейсов AutomationML InterfaceClass и пользовательские классы интерфейсов InterfaceClass хранятся одинаково в формате CAEX InterfaceClass.

- чтобы гарантировать алгоритмическую интерпретируемость семантики пользовательских классов интерфейсов, нужно данные классы выводить из классов интерфейсов AutomationML InterfaceClass.

Рисунок 18 содержит пример пользовательского класса интерфейсов «MyDigitalInput (мой цифровой вход)», полученного из класса интерфейсов AutomationML InterfaceClass «SignalInterface». Соотношения наследования между классом интерфейсов «MyDigitalInput» и стандартным классом интерфейсов AutomationML InterfaceClass «SignalInterface» обеспечивают автоматическую идентификацию данного пользовательского класса как цифрового входного интерфейса. Пользовательские атрибуты заданы надлежащим образом. Пользовательские атрибуты, задействованные в данном примере, не являются предметом рассмотрения настоящего стандарта.

Примечание — В данном примере представление пути доступа сокращено для улучшения читаемости текста. В реальных приложениях пути доступа должны указываться полностью.

7.4 Пользовательский ролевой класс RoleClass

Все классы ролей RoleClass, определенные в МЭК 62714, называются ролевыми классами AutomationML RoleClass. Все классы ролей RoleClass, не определенные в МЭК 62714, называются пользовательскими ролевыми классами RoleClass.

Для пользовательских ролевых классов RoleClass, используются нижеследующие положения:

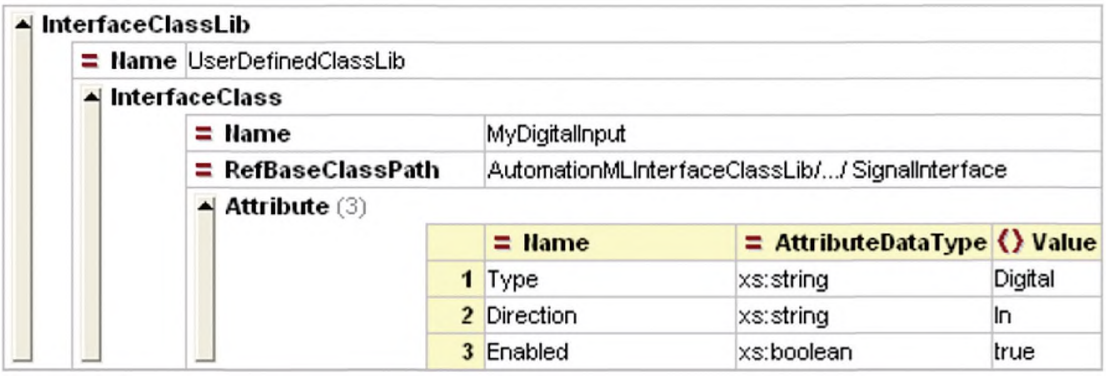
- классы ролей формата CAEX RoleClass должны храниться в соответствии с требованиями формата CAEX RoleClass, определенными в МЭК 62424 (раздел A.2.6);

Примечание 1 — Ролевой класс AutomationML RoleClass и пользовательский ролевой класс RoleClass хранятся одинаково в формате CAEX RoleClass.

- чтобы гарантировать семантическую интерпретируемость пользовательских ролевых классов RoleClass, они должны быть получены из ролевых классов AutomationML RoleClass.

Примечание 2 — Это способствует алгоритмической интерпретируемости семантики класса.

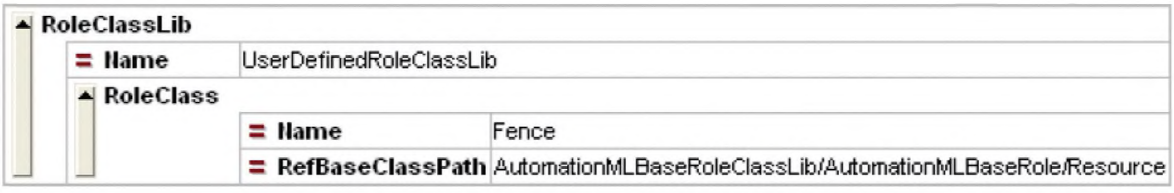
Рисунок 19 содержит пример пользовательского класса «Fence» (ограждение), полученного из стандартного ролевых классов AutomationML RoleClass «Resource». Соотношение наследования между классами «Fence» и «Resource» позволяет интерпретировать данный пользовательский класс в качестве ресурса.




```
<InterfaceClassLib Name="UserDefinedClassLib">
  <InterfaceClass Name="MyDigitalInput" RefBaseClassPath="AutomationMLInterfaceClassLib/.../ SignalInterface">
    <Attribute Name="Type" AttributeDataType="xs:string">
      <Value>Digital</Value>
    </Attribute>
    <Attribute Name="Direction" AttributeDataType="xs:string">
      <Value>In</Value>
    </Attribute>
    <Attribute Name="Enabled" AttributeDataType="xs:boolean">
      <Value>true</Value>
    </Attribute>
  </InterfaceClass>
</InterfaceClassLib>
```

InterfaceClassLib — Библиотека класса интерфейсов; Name — Имя; InterfaceClass — Класс интерфейсов; RefBaseClassPath — Ссылочный путь доступа к базовому классу; Attribute — Атрибут; AttributeDataType — Тип данных атрибута; Digital — Цифровой; xs:string — Строка; xs:boolean — Булевский; true — Истина

Рисунок 18 — Пример представления пользовательского класса интерфейсов InterfaceClass в пользовательской библиотеке InterfaceClassLib




```
<RoleClassLib Name="UserDefinedRoleClassLib">
  <RoleClass Name="Fence" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource"/>
</RoleClassLib>
```

RoleClassLib — Библиотека ролевых классов; Name — Имя; RoleClass — Ролевой класс; RefBaseClassPath — Ссылочный путь доступа к базовому классу

Рисунок 19 — Пример размещения пользовательского класса ролевых классов RoleClass в пользовательской библиотеке RoleClassLib

7.5 Пользовательский класс системных единиц SystemUnitClass

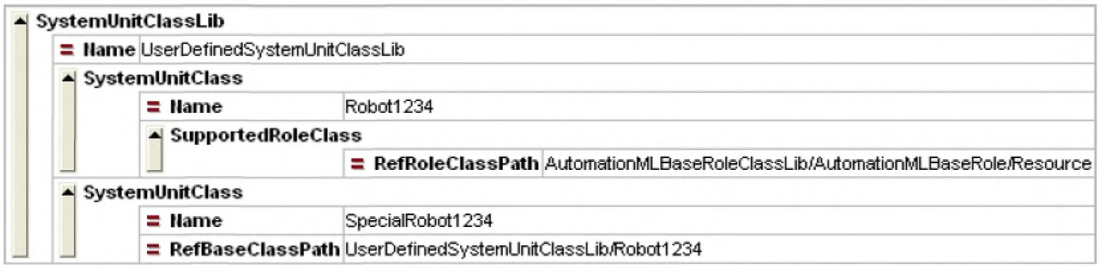
Все классы системных единиц SystemUnitClass являются пользовательскими. В МЭК 62714 классы SystemUnitClass не рассматриваются.

Для пользовательских классов SystemUnitClass, используются нижеследующие положения:

- пользовательские классы SystemUnitClass должны храниться на языке AutomationML в формате CAEX SystemUnitClass. Соответствующее определение приведено в МЭК 62424 (раздел A.2.3);
- пользовательские классы SystemUnitClass должны непосредственно или косвенно быть назначены для ролевых классов AutomationML RoleClass. При необходимости, они используют атрибуты AutomationML;

Пример — На рисунке 20 приведено определение пользовательского класса системных единиц SystemUnitClass с помощью двух различных примеров.

- класс SystemUnitClass «Robot1234» отображает пользовательский класс, поддерживающий роль «Resource» стандартной библиотеки AutomationML RoleClassLib. Данный класс автоматически интерпретируется как «Resource» (ресурс);
- класс SystemUnitClass «SpecialRobot1234» отображает новый пользовательский класс, полученный из предшествующего класса «Robot1234». Таким образом, данный класс также является ресурсом.




```
<SystemUnitClassLib Name="UserDefinedSystemUnitClassLib">
  <SystemUnitClass Name="Robot1234">
    <SupportedRoleClass RefRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource"/>
  </SystemUnitClass>
  <SystemUnitClass Name="SpecialRobot1234" RefBaseClassPath="UserDefinedSystemUnitClassLib/Robot1234"/>
</SystemUnitClassLib>
```

SystemUnitClassLib — Библиотека класса системных единиц; Name — Имя; SystemUnitClass — Класс системных единиц; SupportedRoleClass — Поддерживаемый ролевой класс; RefRoleClassPath — Путь доступа к ссылочному ролевому классу

Рисунок 20 — Примеры для различных пользовательских классов системных единиц SystemUnitClass

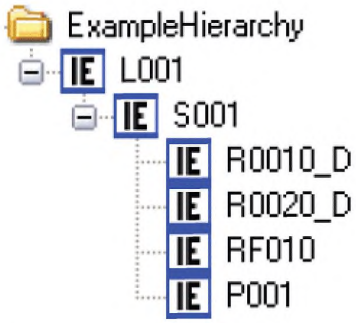
7.6 Пользовательские иерархии экземпляров InstanceHierarchies

Формат иерархий CAEX InstanceHierarchies обеспечивает хранение индивидуальной и проектно-технической информации. Данные иерархии формируют основу формата верхнего уровня языка AutomationML. Они содержат все индивидуальные объекты данных, включая свойства, интерфейсы, соотношения и ссылки.

Для пользовательских иерархий экземпляров InstanceHierarchy, используются нижеследующие положения:

- настоящий стандарт не ограничивает глубину уровней иерархии;
- настоящий стандарт не ограничивает архитектурные правила иерархии;
- настоящий стандарт не определяет конвенции именования рассматриваемых иерархий;
- каждый объект языка AutomationML внутри рассматриваемой иерархии экземпляров InstanceHierarchy должен непосредственно или косвенно закрепляться за ролевым классом AutomationML RoleClass для указания своего абстрактного типа.

Рисунок 21 содержит пример иерархии, включающей линию «L001» со станцией «S001», содержащей двух роботов «R0010_D» и «R0020_D», конвейер «RF010» и программируемый логический контроллер PLC «P001».



ExampleHierarchy — Иерархия примера

Рисунок 21 — Пример пользовательской иерархии экземпляров InstanceHierarchy

На рисунке 22 приведено представление AutomationML для рассматриваемой структуры. В соответствии с МЭК 62424 (раздел А.2.9), каждый объект ассоциирован с ролевым классом RoleClass.

```
<InstanceHierarchy Name="ExampleHierarchy">
  <InternalElement Name="L001" ID="GUID1">
    <InternalElement Name="S001" ID="GUID2">
      <InternalElement Name="R0010_D" ID="GUID3" RefBaseSystemUnitPath="RobotLibrary/Robot_1234">
        <RoleRequirements RefBaseRoleClassPath="AutomationMLRoleClassLib/AutomationMLBaseRole/Resource"/>
      </InternalElement>
      <InternalElement Name="R0020_D" ID="GUID4" RefBaseSystemUnitPath="RobotLibrary/Robot_1234">
        <RoleRequirements RefBaseRoleClassPath="AutomationMLRoleClassLib/AutomationMLBaseRole/Resource"/>
      </InternalElement>
      <InternalElement Name="RF010" ID="GUID5">
        <RoleRequirements RefBaseRoleClassPath="AutomationMLRoleClassLib/AutomationMLBaseRole/Resource"/>
      </InternalElement>
      <InternalElement Name="P001" ID="GUID6">
        <RoleRequirements RefBaseRoleClassPath="AutomationMLCSRoleClassLib/ControlEquipment/ControlHardware/Controller/PLC"/>
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLRoleClassLib/AutomationMLBaseRole/Resource/Structure"/>
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLRoleClassLib/AutomationMLBaseRole/Resource/Structure"/>
  </InternalElement>
</InstanceHierarchy>
```

Рисунок 22 — Представление AutomationML для пользовательской иерархии экземпляров InstanceHierarchy

8 Расширенные понятия языка AutomationML

8.1 Общие положения

В настоящем стандарте определены расширенные понятия для моделирования специальных инженерных аспектов. Справочный обзор и примеры приведены в А.2.

8.2 Объект порт AutomationML Port

Порт AutomationML Port — это объект языка AutomationML, группирующий несколько интерфейсов. Справочный обзор понятий порта и примеры приведены в А.2.2.

Для портов AutomationML Port, используются нижеследующие положения:

- порт AutomationML Port должен быть описан в формате внутренних элементов CAEX InternalElements в ассоциации с ролевым классом RoleClass «Port», описанным в разделе 6.4.5;
- объект порта AutomationML моделируется как дочерний объект рассматриваемого объекта (класса) AutomationML;
- необходимый набор интерфейсов описывается в формате внешних интерфейсов CAEX ExternalInterface объекта порта;

- объект порта не должен содержать дочерних внутренних элементов CAEX InternalElements;
- все внешние интерфейсы CAEX ExternalInterface объектов порта должны непосредственно или косвенно выводиться из класса интерфейсов AutomationML, определенного в разделе 6.3;
- порт AutomationML Port должен дополнительно иметь, по крайней мере, один внешний интерфейс CAEX ExternalInterface, полученный из класса интерфейсов AutomationML InterfaceClass «PortConnector», описанного в разделе 6.3.4.

Примечание — Дополнительные нормативные положения (в части атрибутов объекта порта) приведены в разделе 6.4.5.

8.3 Объект фасет AutomationML Facet

Фасет AutomationML Facet — это объект языка AutomationML, определяющий общее для некоторой совокупности атрибутов (интерфейсов) языка AutomationML представление одного родительского объекта языка AutomationML. Данное понятие облегчает хранение настроек различных конфигураций (например, параметров человеко-машинных интерфейсов HMI, параметров программируемых логических контроллеров PLC) и позволяет автоматизировать некоторые шаги технического управления. Например, настоящий стандарт определяет ролевой класс AutomationML RoleClass «Facet» (см. раздел 6.4.4). Справочный обзор понятия фасет и примеры приведены в разделе А.2.3.

Для фасета AutomationML Facet используются нижеследующие положения:

- объект фасет AutomationML Facet описывается в формате внутренних элементов CAEX InternalElements в ассоциации с ролевым классом RoleClass «Facet», описанным в разделе 6.4.4;
- объект фасет AutomationML Facet моделируется как дочерний объект рассматриваемого объекта (класса) AutomationML;
- фасет должен иметь уникальное среди ближайших родственных элементов произвольное имя;

Примечание — Имя фасета имеет важное значение для ассоциации с понятием Group. Описания понятий и примеры приведены в разделе А.2.3.

- объект (класс) языка AutomationML может иметь произвольное количество объектов фасет;
- фасеты могут иметь произвольное количество атрибутов;
- атрибут фасета должен относиться к существующему атрибуту родительского объекта AutomationML. Идентификатор должен совпадать с именем. Атрибуты фасета, не являющиеся частью родительского объекта, недопустимы;
- интерфейс фасета должен относиться к существующему интерфейсу родительского объекта. Идентификатор должен совпадать с именем. Интерфейсы фасета, не являющиеся частью родительского объекта, недопустимы;
- фасеты не должны содержать новых дочерних объектов, атрибутов, интерфейсов;
- объект фасет не должен быть встроенным;
- фасеты не должны модифицировать существующие атрибуты и интерфейсы.

8.4 Объект группа AutomationML Group

Понятие группы AutomationML Group позволяет отделить структурную информацию от информации об экземпляре. Справочный обзор понятия группы и примеры приведены в разделе А.2.4.

Для объектов группы AutomationML Group используются нижеследующие положения:

- объекты группы AutomationML Group описываются в формате CAEX InternalElements в ассоциации с ролевым классом RoleClass «Group», определенным в разделе 6.4.3;
- объекты группы AutomationML Group моделируются в произвольной позиции иерархии экземпляров InstanceHierarchy или класса системных единиц SystemUnitClass;
- количество объектов группы AutomationML Group не ограничено;
- объект группы AutomationML Group должен содержать только зеркальные объекты и/или последующие объекты группы;

Примечание 1 — Таким образом, объекты группы могут быть встроенными.

Примечание 2 — Если экземпляр А ссылается на другой экземпляр А*, то А называется «зеркальным объектом», а А* — «главным объектом» (в соответствии с МЭК 62424:2008 (раздел А.2.14)). Зеркальный объект ссылается на главный объект и все его данные. Таким образом, зеркальный объект работает как указатель главного объекта.

- группы AutomationML не должны использоваться для описания иерархии установки;

- объект группы AutomationML Group может хранить дополнительную информацию в виде атрибутов, интерфейсов или портов (для описания специальной информации о группе);

Примечание 3 — Указанные дополнительные атрибуты, порты и интерфейсы не идентичны атрибутам, портам или интерфейсам рассматриваемых зеркальных объектов.

- не разрешается изменять существующие атрибуты, интерфейсы или порты зеркальных объектов или дополнять информацию о зеркальных объектах;
- зеркальные объекты должны иметь свой собственный уникальный идентификатор ID;

Примечание 4 — Зеркальный объект считается идентичным главному объекту. Идентификатор помогает отличить зеркальное представление от главного.

- если главный объект стирается, то все соответствующие зеркальные объекты также стираются во избежание неоднозначности;

Примечание 5 — Рассматриваемая функциональность инструментального средства в настоящем стандарте не рассматривается.

- если зеркальный объект стирается, то главный объект не должен быть поврежден;
- если используется атрибут ассоциированного фасета «Associated Facet», то его значение обеспечивает корректность имени существующего фасета.

8.5 Набор свойств AutomationML PropertySet

Набор свойств PropertySet — это ролевой класс, содержащий набор атрибутов с четким синтаксисом и семантикой. Данный набор моделируется как ролевой класс, полученный из стандартного ролевых классов «PropertySet». Концептуальный обзор приведен в разделе А.2.5.

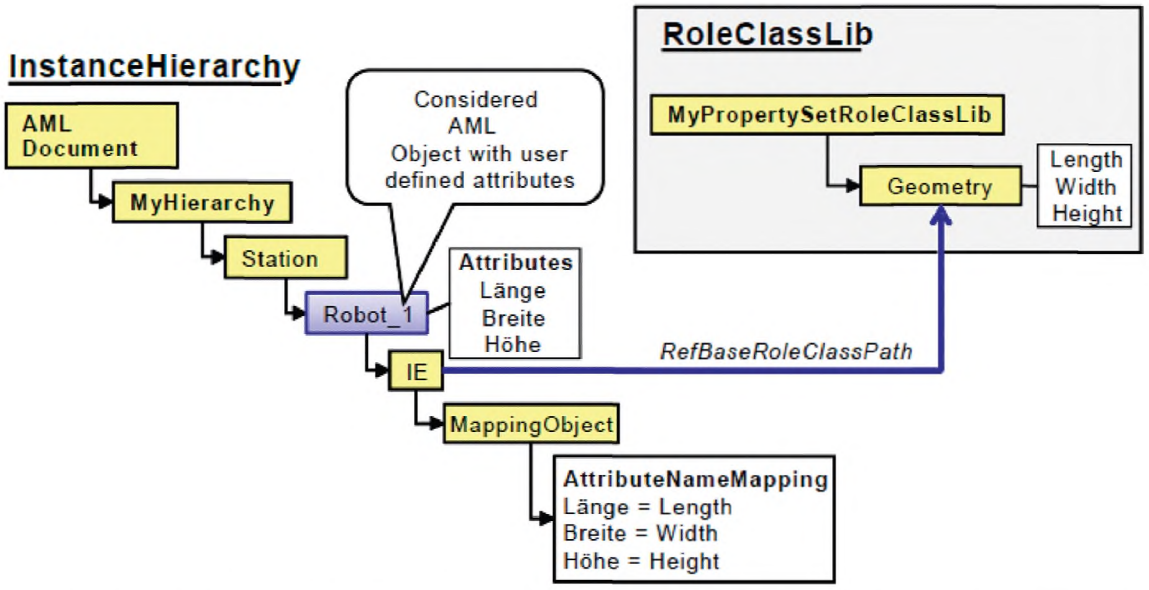
Для понятия набор свойств PropertySet используются нижеследующие положения:

- класс PropertySet моделируется как ролевой класс. Он выводится непосредственно или косвенно из стандартного ролевых классов «PropertySet»;
- классы PropertySet могут собираться в одну или несколько библиотек ролевых классов;
- объекты AutomationML могут ассоциироваться с одним и более классом набора свойств;
- для каждого набора свойств PropertySet объекта AutomationML создаются отдельные дочерние внутренние элементы CAEX InternalElements объекта AutomationML, которые не определяют какие-либо атрибуты CAEX, интерфейсы или внутренние элементы InternalElements, за исключением имени и идентификатора ID. Дочерний объект должен ассоциировать ролевой класс PropertySet с помощью элемента CAEX «RoleRequirement»;
- отображения между атрибутами объекта AutomationML и ролью PropertySet моделируются с помощью элементов отображения CAEX «MappingObject» и «AttributeNameMapping» внутри соответствующих дочерних внутренних элементов InternalElements. Указанные отображения между рассматриваемым объектом AutomationML и ссылочным набором свойств PropertySet являются корректными. Отображенные атрибуты копируются в раздел RoleRequirement. Это копирование не является обязательным;
- атрибуты PropertySet могут быть вложенными;
- ассоциации между объектом AutomationML и несколькими наборами свойств моделируются с помощью нескольких дочерних элементов объекта AutomationML, каждый из которых имеет свою собственную ассоциацию RoleRequirement с соответствующим набором свойств, а также свои собственные отображения.

Рисунок 23 иллюстрирует вышесказанное на примере. Объект Robot_1 имеет несколько пользовательских атрибутов. Дочерние внутренние элементы InternalElements (IE) ассоциируются с геометрией набора свойств PropertySet «Geometry», определяющей соответствующие атрибуты. Объект отображения внутреннего элемента MappingObject IE содержит описание указанного отображения собственных и стандартных атрибутов.

8.6 Поддержка нескольких ролей

В добавление к МЭК 62424 (раздел А.3.18), настоящий стандарт определяет порядок поддержки нескольких ролей для экземпляра объекта. Рассмотрение сразу нескольких ролей имеет смысл, если объект поддерживает несколько различных функциональных возможностей. Например, устройство, которое одновременно является сканером, принтером и факсом. Обзор и примеры приведены в разделе А.2.7.



InstanceHierarchy — Иерархия экземпляров; AutomationML Document — Документ языка AutomationML; MyHierarchy — Моя иерархия; Station — Станция; IE — Внутренний элемент; MappingObject — Объект отображения; AttributeNameMappin: Länge = Length, Breite = Width, Höhe = Height — Отображение имени атрибута: длина, ширина, высота; Considered AutomationML Object with user defined attributes — Рассматриваемый объект языка AutomationML с пользовательскими атрибутами; RoleClassLib — Библиотека ролевых классов; MyPropertySetRoleClassLib — Моя библиотека ролевых классов набора свойств; Geometry — Геометрия; Length, Width, Height — Длина, ширина, высота; Attribute Länge, Breite, Höhe — Атрибуты: длина, ширина, высота; RefBaseRoleClassPath — Ссылочный путь доступа к базовому ролевому классу

Рисунок 23 — Пример, иллюстрирующий понятие набора свойств PropertySet

Для поддержки сразу нескольких ролей используются нижеследующие положения:

- если экземпляр поддерживает только одну роль, то она описывается с помощью атрибута CAEX «RefBaseRoleClassPath», принадлежащего требованию RoleRequirement;

Примечание 1 — Вышесказанное соответствует МЭК 62424 (раздел А.3.18), в котором устанавливается единовременная поддержка только одной роли.

- если экземпляр поддерживает несколько ролей, то каждая из них определяется с помощью элемента CAEX «SupportRoleClass» вместо использования атрибута CAEX для пути доступа к базовому ролевому классу «RefBaseRoleClassPath»;

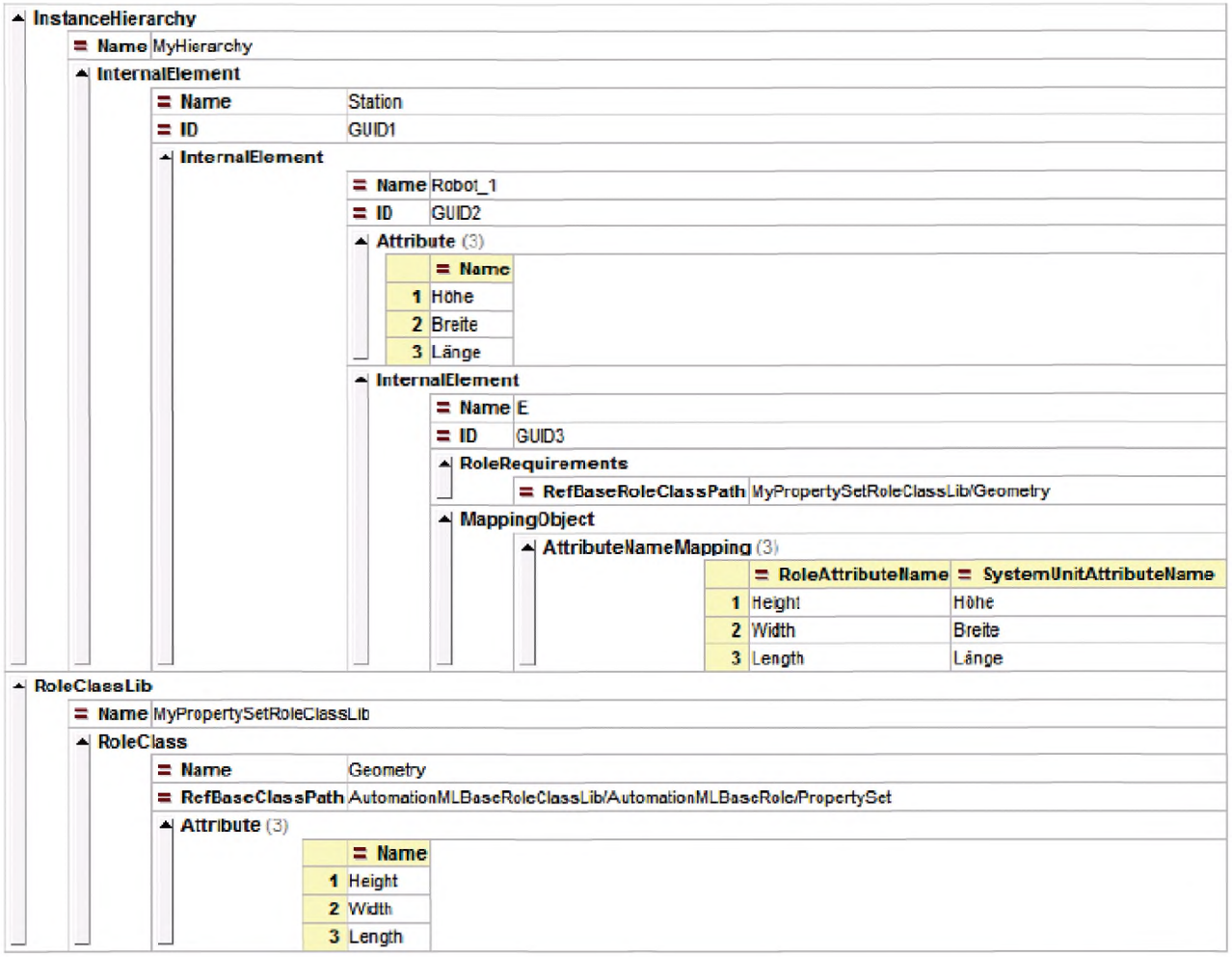
Примечание 2 — Атрибут «RefBaseRoleClassPath» может назначаться только один раз для элемента требования RoleRequirement. При этом элемент CAEX «SupportRoleClass» может быть определен несколько раз. Он является ключевым для назначения нескольких ролей. Однако, в соответствии с МЭК 62424, это слабое семантическое расширение. Оно не изменяет формат данных CAEX.

- если экземпляр поддерживает несколько ролей, то требования к различным ролям должны храниться в самом экземпляре, это осуществляется с помощью элемента CAEX «RoleRequirement». При этом соответствующие атрибуты и интерфейсы назначаются непосредственно, включая имя роли, строку сепаратора «.» и атрибут (имя интерфейса);

Примечание 3 — В соответствии с МЭК 62424 данное семантическое расширение является слабым. Оно не изменяет формат данных CAEX. Отличие МЭК 62424 (раздел А.3.18) заключается в том, что имя роли добавляется в определение атрибута (интерфейса). Пример см. в разделе А.2.7.

- если несколько поддерживаемых ролевых классов уже указаны и элемент CAEX «RoleRequirement» одновременно ассоциирован с определенным путем доступа «RefBaseRoleClass Path», то ассоциированный ролевой класс является предпочтительной ролью. В данном случае определения атрибута и отображения атрибута (интерфейса) требования к роли RoleRequirement (без явного префикса имени роли) ассоциируются с данной предпочтительной ролью.

Примечание 4 — Указанная предпочтительная роль используется в соответствии с МЭК 62424 (приложение А) без семантического расширения.



- InstanceHierarchy
  - Name: MyHierarchy
  - InternalElement
    - Name: Station
    - ID: GUID1
    - InternalElement
      - Name: Robot_1
      - ID: GUID2
      - Attribute (3)

	Name
1	Höhe
2	Breite
3	Länge

      - InternalElement
        - Name: IE
        - ID: GUID3
        - RoleRequirements
          - RefBaseRoleClassPath: MyPropertySetRoleClassLib/Geometry
        - MappingObject
          - AttributeNameMapping (3)

	RoleAttributeName	SystemUnitAttributeName
1	Height	Höhe
2	Width	Breite
3	Length	Länge

- RoleClassLib
  - Name: MyPropertySetRoleClassLib
  - RoleClass
    - Name: Geometry
    - RefBaseClassPath: AutomationMLBaseRoleClassLib/AutomationMLBaseRole/PropertySet
    - Attribute (3)

	Name
1	Height
2	Width
3	Length

```
<InstanceHierarchy Name="MyHierarchy">
  <InternalElement Name="Station" ID="GUID1">
    <InternalElement Name="Robot_1" ID="GUID2">
      <Attribute Name="Höhe"/>
      <Attribute Name="Breite"/>
      <Attribute Name="Länge"/>
      <InternalElement Name="IE" ID="GUID3">
        <RoleRequirements RefBaseRoleClassPath="MyPropertySetRoleClassLib/Geometry"/>
        <MappingObject>
          <AttributeNameMapping RoleAttributeName="Height" SystemUnitAttributeName="Höhe"/>
          <AttributeNameMapping RoleAttributeName="Width" SystemUnitAttributeName="Breite"/>
          <AttributeNameMapping RoleAttributeName="Length" SystemUnitAttributeName="Länge"/>
        </MappingObject>
      </InternalElement>
    </InternalElement>
  </InternalElement>
</InstanceHierarchy>
<RoleClassLib Name="MyPropertySetRoleClassLib">
  <RoleClass Name="Geometry" RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/PropertySet">
    <Attribute Name="Height"/>
    <Attribute Name="Width"/>
    <Attribute Name="Length"/>
  </RoleClass>
</RoleClassLib>
```

Рисунок 24 — Пример описания набора свойств PropertySet на языке XML

8.7 Разбиение данных AutomationML верхнего уровня на различные документы

В соответствии с МЭК 62424:2008 (раздел А.2.12), формат CAEX явно поддерживает распределение инженерных данных между различными файлами и обеспечивает работу механизмов ссылок на внешние файлы CAEX с помощью внешней ссылки CAEX «ExternalReference» и соответствующего понятия с косвенным доступом в формате CAEX.

8.8 Интернационализация

Различные языки (для конкретных имен и описаний) могут храниться в языке AutomationML в соответствии со спецификациями языка XML и текстовым форматом UTF-8.

8.9 Информация о версии объекта AutomationML

Для хранения информации о версии и пересмотре индивидуальных объектов AutomationML (экземпляров объектов) используются стандартные версии и утвержденный порядок пересмотра в соответствии с МЭК 62424 (раздел А.2.2.2).

Для хранения информации о версии объектов AutomationML и информации о библиотечной версии AutomationML, см. раздел 5.3.

Хранение специальной метаинформации об инструменте — см. раздел 5.4.

Приложение А
(справочное)

Введение в язык AutomationML

А.1 Общие понятия языка AutomationML

А.1.1 Архитектура языка AutomationML

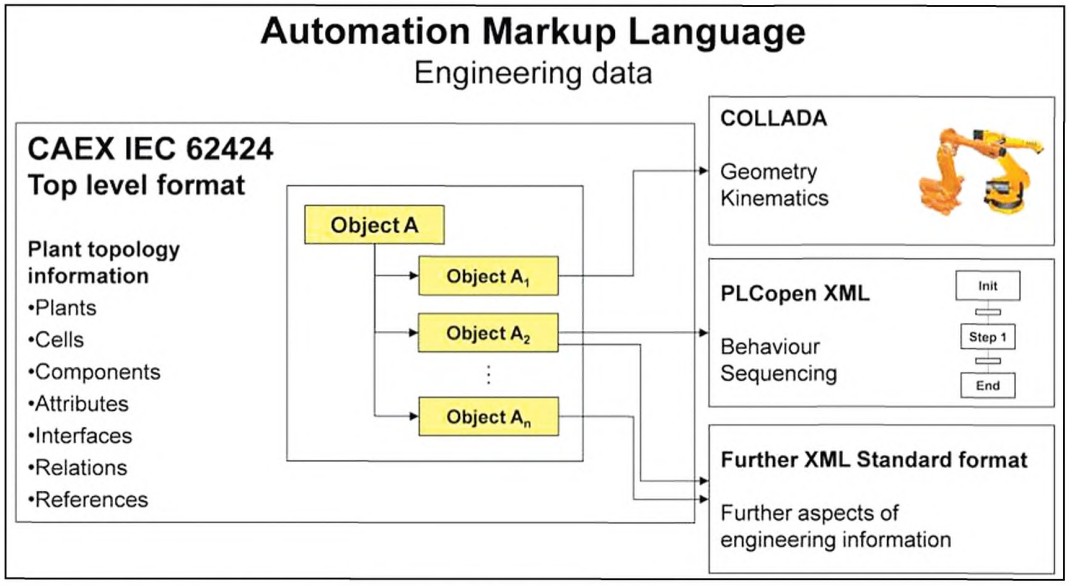
Язык разметки автоматизации AutomationML — это схемный формат данных расширяемого языка разметки XML, разработанный для обеспечения независимого обмена (продавцом) инженерной информации о производственной установке. Целью языка AutomationML является обеспечение взаимосвязи средств инженерии (на неоднородном множестве инструментальных средств инженерии) в различных дисциплинах: проектирование механизированного оборудования, электротехническое проектирование, проектирование производственных процессов, управление производственными процессами, разработка человеко-машинного интерфейса (HMI), программирование логического контроллера (PLC), программирование роботов и т. д.

Язык AutomationML накапливает инженерную информацию в соответствии с имеющейся объектно-ориентированной парадигмой. Он позволяет моделировать конкретные компоненты производственной установки как объекты данных, инкапсулирующие различные технические аспекты. Рассматриваемый объект может включать подобъекты. Он может сам быть частью большой композиции (агрегации). Данный объект может описывать: сигналы, контроллеры PLC, накопители, клапаны управления, роботы, производственные ячейки (с различным уровнем детализации), производственные площадки и производственные линии. Типовые объекты автоматизации производственных установок включают информацию о топологии, геометрии, кинематике и логике. При этом логика включает последовательность/упорядоченность, поведение и управление.

Язык AutomationML объединяет существующие промышленные форматы данных, разработанные для хранения и обмена инженерной информации по различным аспектам. Указанные форматы данных используются «как есть» на основе их собственных спецификаций. Они не подстраиваются под требования языка AutomationML.

Ядром языка AutomationML является формат данных CAEX верхнего уровня. Он соединяет различные форматы данных. Таким образом, язык AutomationML имеет унаследованную распределенную архитектуру документов.

Рисунок А.1 иллюстрирует базовую архитектуру AutomationML, а также распределение информации о топологии, геометрии, кинематике и логике.



Automation Markup language — Язык разметки автоматизации; Engineering data — Инженерные данные; CAEX IEC 62424 Top level format — Формат верхнего уровня CAEX МЭК 62424; Plant topology information — Информация о топологии производственной установки; Plants — Установки; Cells — Ячейки; Components — Компоненты; Attributes — Атрибуты; Interfaces — Интерфейсы; Relations — Соотношения; References — Ссылки; Object A — Объект А; COLLADA — Формат COLLADA; Geometry — Геометрия; Kinematics — Кинематика; PLCopen XML — Формат PLCopen XML; Behavior — Поведение; Sequencing — Упорядоченность; Further XML Standard format — Прочие стандартные форматы XML; Further aspects of engineering information — Прочие аспекты инженерной информации

Рисунок А.1 — Общая архитектура языка AutomationML

Основными преимуществами понятия распределенного документа являются использование опробованных и утвержденных форматов данных, распределение данных по различным файлам, что облегчает обработку массивов информации и упрощает использование библиотечных файлов AutomationML, которые могут отдельно храниться, обмениваться и рассматриваться. Информация с различным уровнем детализации (например, геометрические варианты) может храниться отдельно. Язык AutomationML определяет ассоциации между ссылочными форматами данных и инженерными объектами.

Краткие примеры, приведенные в разделе А.1.1, содержат общий обзор информации, которая может храниться и обмениваться на основе языка AutomationML.

- **Информация о топологии установки:** топология производственной установки описывает установку как иерархическую структуру, включающую индивидуальные объекты установки, представленные индивидуальными объектами данных. Структура данных объектов моделируется на определенном уровне детализации (например, робот целиком, механизм захвата целиком, но не отдельные его валы или соединительные элементы). Данные объекты включают свойства и соотношения с другими объектами в рамках рассматриваемой иерархической структуры. Топология установки работает как структура данных верхнего уровня. Информация хранится в формате данных CAEX в соответствии с МЭК 62424 (раздел 7, приложения А и С). В расширенной версии МЭК 62424, язык AutomationML определяет ссылки из объектов CAEX на информацию, хранящуюся во внешних документах (вне формата CAEX). Подраздел А.1.2 содержит примеры моделирования информации о топологии установки на языке AutomationML.

- **Информация о геометрии и кинематике:** геометрия отдельного объекта установки включает его геометрическое представление. Информация о кинематике описывает физические соединения трехмерных твердых тел и зависимости между объектами. Информация о геометрии и кинематике хранится в файловом формате COLLADA. Дополнительно файл COLLADA включает определение информации о сопряжении геометрии и кинематики. Интерфейсы COLLADA могут быть опубликованы как внешние интерфейсы CAEX ExternalInterface в формате верхнего уровня (для последующего установления взаимосвязей). Полное представление получается автоматически из информации о геометрии в формате COLLADA для различных объектов. На указанные файлы можно ссылаться из формата CAEX. Взаимосвязь указанных файлов обеспечивается с помощью связующих CAEX-механизмов. Краткий пример имеется в подразделе А.1.3 Подробности приведены в МЭК 62714-3.

- **Информация о логике:** информация о логике описывает последовательности действий и поведение объектов, соединения типа ВВОД/ВЫВОД и логические переменные. Данные последовательности описываются и хранятся во внешних документах языка PLCopen XML. Переменные и сигналы публикуются как внешние интерфейсы CAEX ExternalInterface. Указанные документы могут быть ссылочными из среды CAEX. Они могут быть взаимосвязанными элементами внутри самой среды CAEX. Подраздел А.1.4 содержит краткое введение в основные понятия. Подробности приведены в МЭК 62714-4.

- **Информация о ссылках и соотношениях:** язык AutomationML проводит различия между ссылками и соотношениями. Ссылки отображают связи объектов CAEX с информацией, хранящейся вовне. Соотношения отображают ассоциации между объектами CAEX. Более того, для хранения ассоциаций между информацией, хранящейся во внешних документах, используется один и тот же механизм. Кроме того, необходимо опубликовать рассматриваемые связующие элементы с помощью внешнего интерфейса CAEX ExternalInterface в топологии установки CAEX. Подробности выполнения ссылок на документы COLLADA и документы PLCopen XML см. в МЭК 62714-3 и МЭК 62714-4. Подраздел А.1.5 содержит справочный обзор методик моделирования ссылок и соотношений на языке AutomationML. Нормативные положения рассмотрены в подразделах 5.6 и 5.7.

- **Выполнение ссылок на другие форматы данных:** МЭК 62714 может быть расширен в будущем с помощью дополнительных частей, описывающих интеграцию дополнительных форматов данных, использующих механизмы ссылок языка AutomationML.

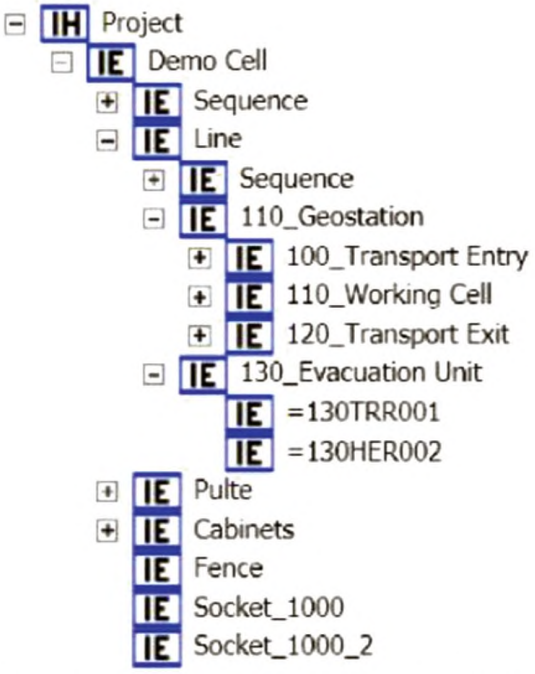
Обмен инженерной информацией требует дополнительных расширенных понятий. Раздел А.2 разъясняет указанные понятия. Раздел 8 содержит нормативные положения.

Примечание — В настоящем стандарте пути доступа в ряде случаев приведены в сокращенной форме. Например, вместо пути доступа «AutomationMLInterfaceClassLib/AutomationMLBaseInterface/Port» указан путь доступа «AutomationMLInterfaceClassLib/.../Port». Это облегчает чтение документа. В реальных документах XML все пути доступа указываются в соответствии с настоящим стандартом.

А.1.2 Моделирование информации о топологии установки

В языке AutomationML конкретные компоненты производственной установки моделируются как объекты данных, инкапсулирующие различные аспекты инженерной информации. Здесь необходимо структурировать объекты данных. Правильный способ структурирования таких объектов данных — построение иерархии объектов, составляющих топологию установки (см. раздел 3.1.20).

Для хранения иерархической структуры установки, язык AutomationML использует понятия в формате данных CAEX верхнего уровня в соответствии с МЭК 62424 (раздел А.2.11). Рисунок А.2 содержит пример топологии промышленной установки производственной линии, содержащей несколько объектов на различных иерархических уровнях.



IH — Внутренняя иерархия; IE — Внутренний элемент; Project — Проект; Demo Cell — Демонстрационная ячейка; Sequence — Последовательность; Line — Линия; Geostation — Геостанция; Transport Entry — Транспортный вход; Working Cell — Рабочая ячейка; Transport Exit — Транспортный выход; Evacuation Unit — Эвакуационный блок; Pulte — Пульт; Cabinets — Кабинеты; Fence — Ограждение; Socket — Разъем

Рисунок А.2 — Представление топологии установки на языке AutomationML

С помощью стандартных понятий CAEX можно моделировать множественные иерархии, пересеченные структуры и сложные сети объектов. При этом ключом к повышению эффективности рассматриваемой объектно-ориентированной парадигмы является доступность библиотек, содержащих предварительно определенные и утвержденные инженерные решения. Кроме того, формат CAEX реализует несколько различных типов библиотек AutomationML для интерфейсов, ролей, системных модулей и иерархий экземпляров, представляющих особую важность для объектов AutomationML.

- **Классы интерфейсов InterfaceClass и библиотеки классов интерфейсов InterfaceClassLib:** интерфейсы способствуют определению соотношений между объектами AutomationML. Подраздел 6.3 содержит описание стандартной библиотеки AML-InterfaceClassLib на множестве абстрактных классов интерфейсов InterfaceClass, применяемых в общих системах автоматизации. Указанные классы включают их синтаксическое и семантическое определение. Они содержат спецификацию пользовательских интерфейсов объектов. Подраздел 7.3 содержит описание методики моделирования пользовательских ролевых классов.

- **Классы ролей RoleClass и библиотеки ролевых классов RoleClassLib:** классы ролей RoleClass обеспечивают определение абстрактных характеристик объектов CAEX. Они обеспечивают автоматическую интерпретацию семантики пользовательских объектов AutomationML. Подраздел 6.4 содержит описание библиотеки ролевых классов AML-RoleClassLib с набором абстрактных ролевых классов RoleClass для общих систем автоматизации. Подраздел 7.4 содержит описание методики моделирования пользовательских ролевых классов. Он также содержит описание последующих библиотек ролей в соответствии с МЭК 62714-2.

- **Системные единицы SystemUnits и библиотека класса системных единиц SystemUnitClassLib:** библиотека класса системных единиц SystemUnitClassLib используется для хранения специальных классов AutomationML продавца. Подраздел 7.5 содержит описание архитектурных правил определения класса системных единиц SystemUnitClass. Язык AutomationML не устанавливает предварительных определений элементов библиотеки SystemUnitClassLib или класса SystemUnitClass.

- **Экземпляры и иерархия экземпляров InstanceHierarchy:** иерархии экземпляров хранят данные топологии текущих проектов и, таким образом, корневых объектов языка AutomationML. Данные иерархии включают экземпляры объектов AutomationML. Подраздел 7.6 содержит описание порядка хранения инженерной информации с помощью иерархии экземпляров типа InstanceHierarchy.

Важным аспектом моделирования топологии производственной установки является идентификация объектов. Разные инженерные инструменты используют разные понятия для идентификации объектов, например, уникальное имя, уникальный идентификатор, уникальный путь доступа. Одни инструменты позволяют изменять иден-

тификаторы в течение срока существования, другие — нет. Внутри одного инструмента это работает отлично. При этом обмен объектов между различными инструментами невозможен. Раздел 5.5 вводит понятие обязательной идентификации объекта. Данное понятие обеспечивает возможность обмениваться данными между различными средствами инженерии с помощью понятия индивидуальной идентификации объекта.

А.1.3 Ссылочная информация о геометрии и кинематике

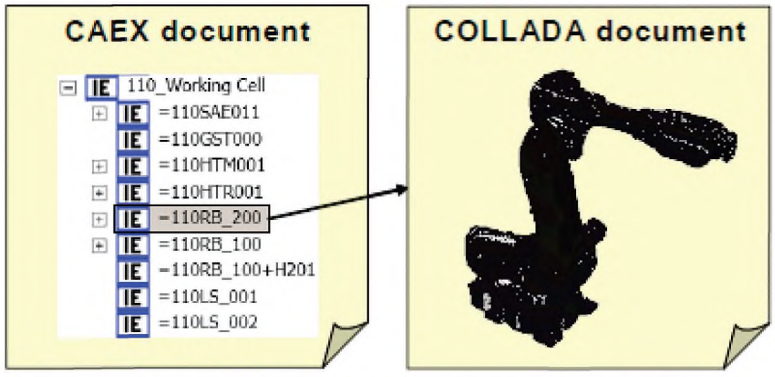
Информация о геометрии и кинематике хранится в отдельных документах в формате данных COLLADA. Моделирование информации о геометрии и кинематике, таким образом, разбивается на две части. С одной стороны, соответствующий объект моделируется внутри формата CAEX без учета какой-либо информации о геометрии и кинематике (в соответствии с настоящим стандартом). С другой стороны, документы COLLADA должны содержать информацию о геометрии и кинематике. Окончательно получается, что объект CAEX содержит ссылку на документ COLLADA в соответствии с МЭК 62714-3.

Рисунок А.3 содержит пример документа AutomationML, включающего объект «110RB_200». Данный объект ссылается на внешний документ COLLADA, содержащий соответствующую информацию о геометрии и кинематике.

Ссылка моделируется с помощью интерфейса CAEX, полученного из стандартного класса интерфейсов языка AutomationML «COLLADAInterface» (см. разделы 5.7 и 6.3.7). Подробности приведены в МЭК 62714-3.

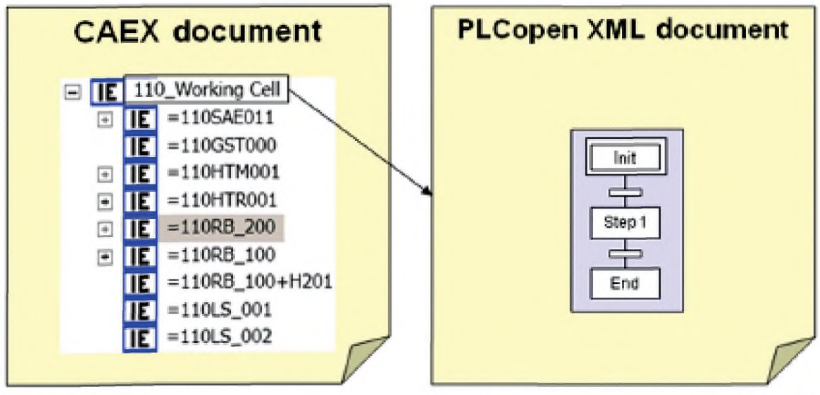
А.1.4 Ссылочная информация о логике

Информация о логике хранится в отдельных документах в формате данных PLCopen XML. Моделирование информации о логике, таким образом, делится на две части. С одной стороны, соответствующие объекты моделируются внутри CAEX без учета какой-либо информации о логике в соответствии с настоящим стандартом. С другой стороны, документ PLCopen XML должен содержать информацию о логике в соответствии с МЭК 62714-4. Окончательно получается, что объект CAEX должен иметь ссылку на документ PLCopen XML. Рисунок А.4 содержит пример документа AutomationML, включающего объект «110_Working Cell» (рабочая ячейка), имеющий ссылку на внешний документ PLCopen XML, содержащий соответствующую информацию о логике.



CAEX document — Документ CAEX; COLLADA document — Документ COLLADA; Working Cell — Рабочая ячейка; E — Внутренний элемент

Рисунок А.3 — Ссылка из объекта CAEX на документ COLLADA



CAEX document — Документ CAEX; PLCopen XML document — Документ PLCopen XML; Working Cell — Рабочая ячейка; Step 1 — Шаг 1; End — Конец; Init — Инициализация

Рисунок А.4 — Ссылка из объекта CAEX на документ PLCopen XML

А.1.5 Моделирование соотношений

Для моделирования объектов необходимо уметь поставить один объект в соответствие другому. Требуется наличие дополнительных механизмов, чтобы связать указанные объекты с данными, хранящимися вовне.

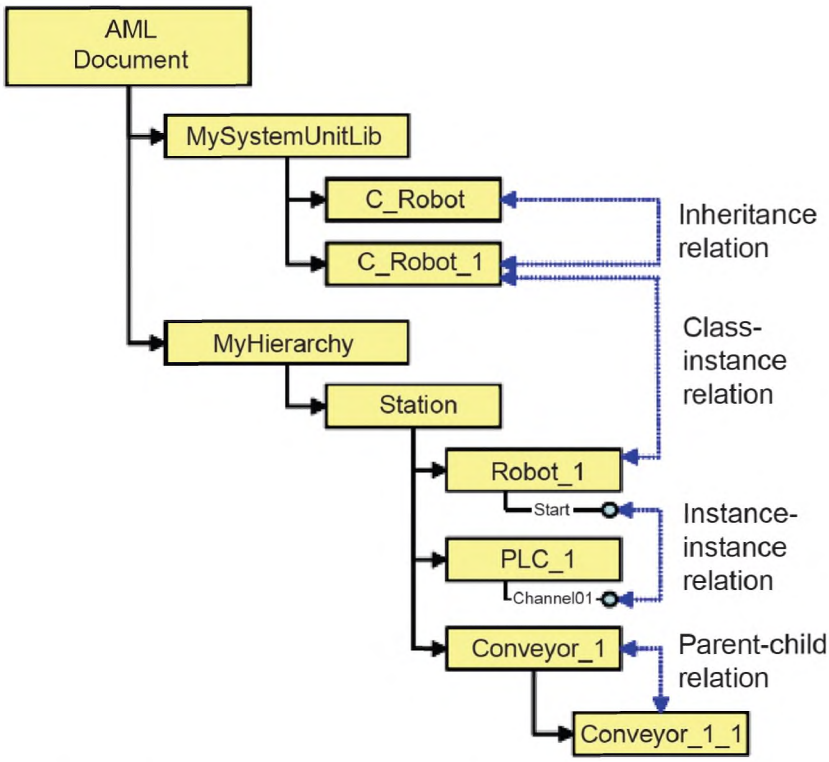
Соотношение выражает ассоциацию между двумя и более объектами. Данная зависимость может быть любой природы (физической или логической). Язык AutomationML поддерживает нижеследующие соотношения:

- соотношения «родитель-потомок» (см. разделы 5.6.2 и 5.6.3):
  - соотношения «родитель-потомок» между объектами AutomationML;
  - соотношения «родитель-потомок» между классами AutomationML;
- соотношения наследования (см. раздел 5.6.4):
  - соотношения наследования между классами системных единиц SystemUnitClass;
  - соотношения наследования между ролевыми классами RoleClass;
  - соотношения наследования между классами интерфейсов InterfaceClass;
- соотношения «класс-экземпляр класса» (см. раздел 5.6.5):
  - соотношения между классом SystemUnitClass и его экземпляром;
  - соотношения между классом RoleClass и его экземпляром;
  - соотношения между классом InterfaceClass и его экземпляром;
- соотношения «экземпляр-экземпляр» (см. раздел 5.6.6):
  - соотношения между объектами AutomationML;
  - соотношения между опубликованными данными, хранящимися вовне.

Рисунок А.5 содержит пример типов соотношений, поддерживаемых языком AutomationML.

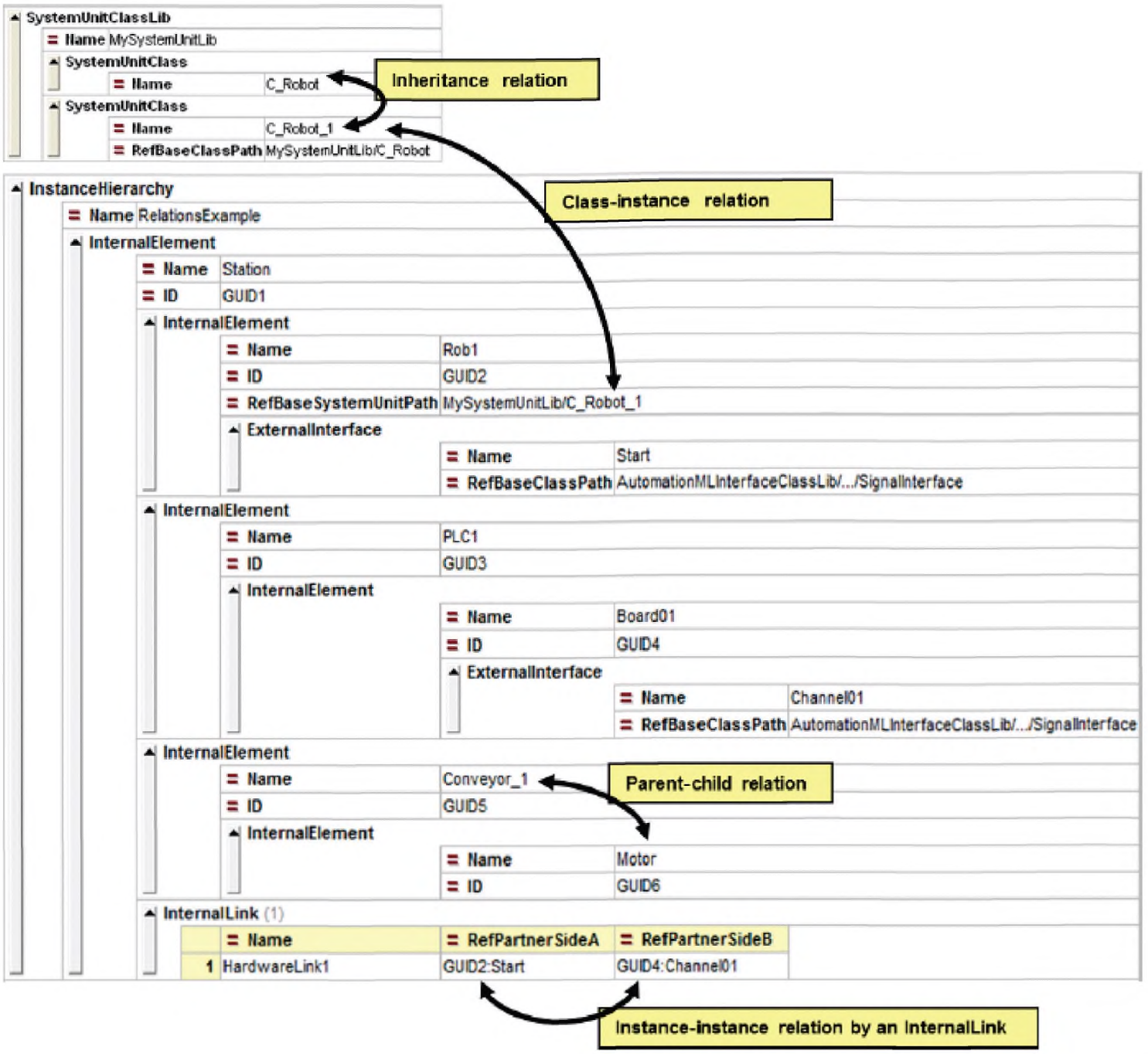
Рисунок А.6 иллюстрирует модель AutomationML, соответствующую данному примеру, с помощью табличного представления. Отметим, что информация о пути доступа может сокращаться с помощью поля «/.../», что улучшает читаемость. Рисунок А.7 содержит описание на языке XML, соответствующее рассматриваемой библиотеке AutomationML.

Рисунок А.8 содержит описание иерархии экземпляров InstanceHierarchy на языке XML.



AutomationML Document — Документ AutomationML; MySystemUnitLib — Моя библиотека системных единиц; C_Robot — Робот С; Inheritance relation — Соотношение наследования; MyHierarchy — Моя иерархия; Station — Станция; Class — instance relation — Соотношение «класс—экземпляр класса»; PLC_1 — Программируемый логический контроллер 1; Start — Старт; Channel01+ — Канал 01+; Instance — Instance relation — Соотношение «экземпляр—экземпляр»; Conveyor_1 — Конвейер 1; Parent — child relation — Соотношение «родитель—потомок»

Рисунок А.5 — Соотношения, поддерживаемые языком AutomationML



SystemUnitClassLib — Библиотека класса системных единиц; Name — Имя; SystemUnitClass — Класс системных единиц; Inheritance relation — Соотношение наследования; RefBaseClassPath — Ссылочный путь доступа к базовому классу; InstanceHierarchy — Иерархия экземпляров; InternalElement — Внутренний элемент; Class-instance relation — Соотношение «класс-экземпляр класса»; ID — Идентификатор; RefBaseSystemUnitPath — Ссылочный путь доступа к базовым системным единицам; ExternalInterface — Внешний интерфейс; Parent-child relation — Соотношение «родитель—потомок»; InternalLink — Внутренняя связь; HardwareLink1 — Аппаратная связь 1; RefPartnerSideA — Ссылочная сторона А партнера; Instance-instance relation by an InternalLink — Соотношение «экземпляр—экземпляр», установленное внутренней связью

Рисунок А.6 — Пример описания соотношений на языке XML

```
<SystemUnitClassLib Name="MySystemUnitLib">
    <SystemUnitClass Name="C_Robot"/>
    <SystemUnitClass Name="C_Robot_1" RefBaseClassPath="MySystemUnitLib/C_Robot"/>
</SystemUnitClassLib>
```

Рисунок А.7 — Описание примера соотношений библиотеки класса системных единиц SystemUnitClassLib на языке XML

```
<InstanceHierarchy Name="RelationsExample">
  <InternalElement Name="Station" ID="GUID1">
    <InternalElement Name="Rob1" ID="GUID2" RefBaseSystemUnitPath="MySystemUnitLib/C_Robot_1">
      <ExternalInterface Name="Start" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
    </InternalElement>
    <InternalElement Name="PLC1" ID="GUID3">
      <InternalElement Name="Board01" ID="GUID4">
        <ExternalInterface Name="Channel01" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
      </InternalElement>
    </InternalElement>
    <InternalElement Name="Conveyor_1" ID="GUID5">
      <InternalElement Name="Motor" ID="GUID6"/>
    </InternalElement>
    <InternalLink Name="HardwareLink1" RefPartnerSideA="GUID2:Start" RefPartnerSideB="GUID4:Channel01"/>
  </InternalElement>
```

Рисунок А.8 — Описание примера соотношений иерархии экземпляров InstanceHierarchy на языке XML

А.2 Расширенные понятия языка AutomationML и примеры

А.2.1 Общий обзор

Язык AutomationML определяет расширенные понятия для моделирования специальных инженерных аспектов. Например, понятие порта AutomationML Port, понятие фасета AutomationML Facet, понятие группы AutomationML Group. В таблице А.1 приведен обзор указанных расширенных понятий.

Таблица А.1 — Обзор основных расширенных понятий языка AutomationML

Понятие	Описание
AutomationML Port	Понятие порта содержит описание высокого уровня для комплексного интерфейса. Понятие AutomationML Port состоит из набора интерфейсов AutomationML, принадлежащих друг другу. Они могут применяться по принципу «вилки» или «розетки»
AutomationML Facet	Фасеты AutomationML Facet позволяют хранить подмножества атрибутов и интерфейсов объекта AutomationML. Они могут рассматриваться как представления конкретных аспектов инженерных данных
AutomationML Group	Понятие AutomationML Group позволяет хранить различные представления подмножества объектов AutomationML. Оно может служить фильтром объектов для древа производственной установки для различных средств инженерии
PropertySet	Понятие набора свойств PropertySet обеспечивает отображение собственных атрибутов пользовательских объектов AutomationML на предварительно семантически определенные атрибуты. Указанные семантически согласованные атрибуты хранятся в классах ролей PropertySet
Process-Product-Resource	Понятие «Процесс-Продукт-Ресурс» обеспечивает высокий уровень структурирования инженерных данных, основанных на процесс-ориентированном (продукт-ориентированном, ресурс-ориентированном) представлении, включающем соотношения между ними

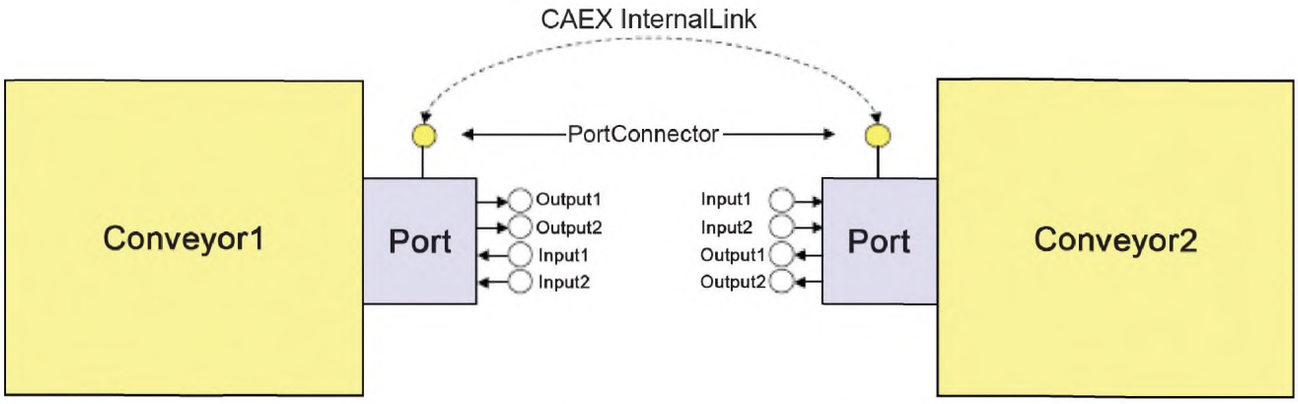
А.2.2 Понятие порта AutomationML Port

А.2.2.1 Описание понятия порт

Порт AutomationML Port — это объект языка AutomationML, группирующий несколько интерфейсов (см. рисунок А.9). Объект порта принадлежит одному родительскому объекту языка AutomationML и описывает комплексные интерфейсы родительского объекта. Порты соединяются друг с другом на более высоком уровне абстракции, чем просто связь с каждым отдельным интерфейсом. Понятие порта AutomationML Port может описывать «вилки», «розетки» и любые другие группы интерфейсов, непосредственно соединенные друг с другом. Кроме того, язык AutomationML определяет понятие AutomationML для ролевых классов RoleClass «Port» (см. раздел 6.4.5). Нормативные положения рассмотрены в разделе 8.2.

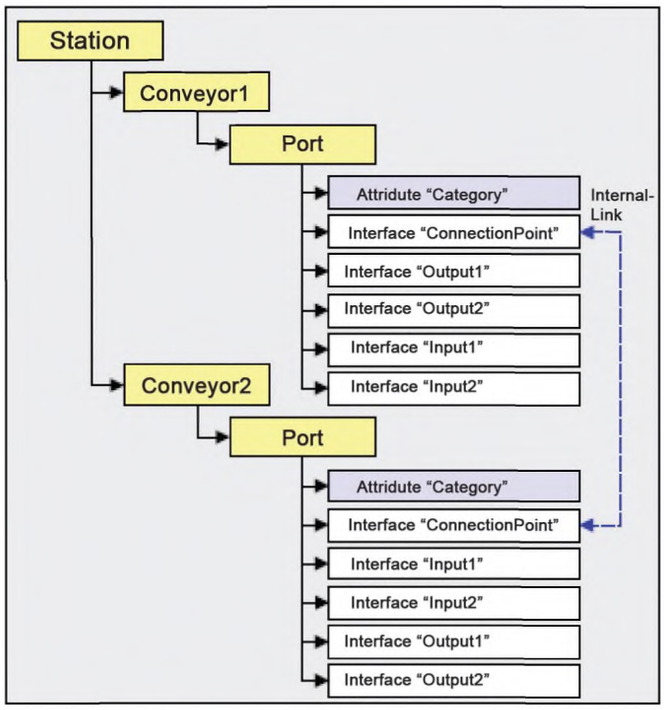
А.2.2.2 Пример

Рисунок А.10 содержит пример понятия порта AutomationML Port. Объект «Station» (станция) включает подобъекты «Конвейер1» и «Конвейер2». Оба подобъекта имеют один объект порта каждый. Объект порта содержит набор интерфейсов, а также стандартный интерфейс «Connection-Point» (точка соединения), полученный из клас-



CAEX InternalLink — Внутренняя связь CAEX; Conveyor1 — Конвейер 1; Port — Порт; PortConnector — Коннектор порта; Output1 — Выход 1; Input2 — Вход 2

Рисунок А.9 — Понятие порта



Station — Станция; Conveyor1 — Конвейер 1; Port — Порт; Attribute Category — Атрибут «категория»; Interface «ConnectionPoint» — Интерфейс «точка соединения»; Interface «Output 1» — Интерфейс «выход 1»; Interface «Input1» — Интерфейс «вход 1»; Internal Link — Внутренняя связь

Рисунок А.10 — Пример описания понятия AutomationML Port

са интерфейсов AutomationML InterfaceClass «PortConnector». Данный стандартный интерфейс может соединяться с помощью внутренних связей CAEX InternalLinks. Рассматриваемое соотношение означает, что оба порта соединены друг с другом. Внутренние связи субинтерфейсов не описаны здесь в подробностях. Рассматривается соединение только абстрактных точек ConnectionPoints. Кроме данного понятия, язык AutomationML допускает хранение каждого индивидуального связующего элемента, расположенного между субинтерфейсами.

Рисунки А.11 и А.12 описывают практическую реализацию на языке AutomationML примера системы, приведенной на рисунке А.10.

А.2.2.3 Моделирование порта как пользовательского класса системных единиц AutomationML SystemUnitClass

Рисунок А.13 содержит пример описания пользовательского класса системных единиц SystemUnitClass «myPortClass» на языке XML.

- InstanceHierarchy
  - Name: PortExample
  - InternalElement
    - Name: Station
    - ID: GUID1
    - InternalElement
      - Name: Conveyor1
      - ID: GUD2
      - InternalElement
        - Name: Port
        - ID: GUID3
        - Attribute
          - Name: Direction
          - AttributeDataType: xs:string
          - Value: Out
        - ExternalInterface (5)

	Name	RefBaseClassPath
1	ConnectionPoint	AutomationMLInterfaceClassLib/.../PortConnector
2	Output1	AutomationMLInterfaceClassLib/.../SignalInterface
3	Output2	AutomationMLInterfaceClassLib/.../SignalInterface
4	Input1	AutomationMLInterfaceClassLib/.../SignalInterface
5	Input2	AutomationMLInterfaceClassLib/.../SignalInterface

        - RoleRequirements
          - RefBaseRoleClassPath: AutomationMLBaseRoleClassLib/.../Port
      - RoleRequirements
        - RefBaseRoleClassPath: AutomationMLBaseRoleClassLib/.../Resource
    - InternalElement
      - Name: Conveyor2
      - ID: GUD4
      - InternalElement
        - Name: Port
        - ID: GUID5
        - Attribute
          - Name: Direction
          - AttributeDataType: xs:string
          - Value: In
        - ExternalInterface (5)

	Name	RefBaseClassPath
1	ConnectionPoint	AutomationMLInterfaceClassLib/.../PortConnector
2	Input1	AutomationMLInterfaceClassLib/.../SignalInterface
3	Input2	AutomationMLInterfaceClassLib/.../SignalInterface
4	Output1	AutomationMLInterfaceClassLib/.../SignalInterface
5	Output2	AutomationMLInterfaceClassLib/.../SignalInterface

        - RoleRequirements
          - RefBaseRoleClassPath: AutomationMLBaseRoleClassLib/.../Port
      - RoleRequirements
        - RefBaseRoleClassPath: AutomationMLBaseRoleClassLib/.../Resource
    - InternalLink (1)

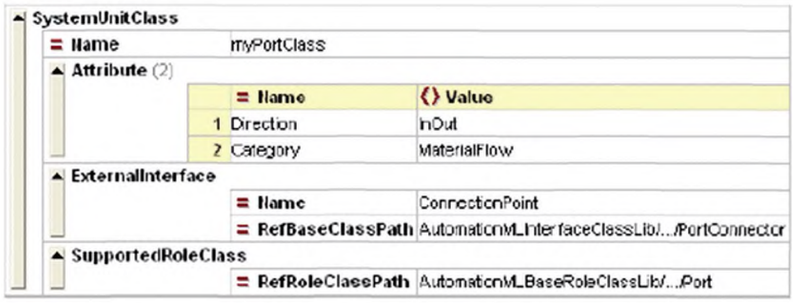
	Name	RefPartnerSideA	RefPartnerSideB
1	L1	GUD3:ConnectionPoint	GUD5:ConnectionPoint

InstenceHierarchy — Иерархия экземпляров; Name — Имя; InternalElement — Внутренний элемент; ID — Идентификатор; Attribute — Атрибут; AttributeDataType — Тип данных атрибута; Value — Значение; ExternalInterface — Внешний интерфейс; RefBaseClassPath — Ссылочный путь доступа к базовому классу; RoleRequirements — Требования роли; RefBaseRoleClassPath — Ссылочный путь доступа к базовому ролевому классу; ConnectionPoint — Точка соединения; RefPartnerSideA — Ссылочная сторона А партнера; InternalLink — Внутренняя связь

Рисунок А.11 — Описание понятия порта AutomationML Port на языке XML

```
<InstanceHierarchy Name="PortExample">
  <InternalElement Name="Station" ID="GUID1">
    <InternalElement Name="Conveyor1" ID="GUID2">
      <InternalElement Name="Port" ID="GUID3">
        <Attribute Name="Direction" AttributeDataType="xs:string">
          <Value>Out</Value>
        </Attribute>
        <ExternalInterface Name="ConnectionPoint" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PortConnector"/>
        <ExternalInterface Name="Output1" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
        <ExternalInterface Name="Output2" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
        <ExternalInterface Name="Input1" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
        <ExternalInterface Name="Input2" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Port"/>
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Resource"/>
    </InternalElement>
    <InternalElement Name="Conveyor2" ID="GUID4">
      <InternalElement Name="Port" ID="GUID5">
        <Attribute Name="Direction" AttributeDataType="xs:string">
          <Value>In</Value>
        </Attribute>
        <ExternalInterface Name="ConnectionPoint" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PortConnector"/>
        <ExternalInterface Name="Input1" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
        <ExternalInterface Name="Input2" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
        <ExternalInterface Name="Output1" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
        <ExternalInterface Name="Output2" RefBaseClassPath="AutomationMLInterfaceClassLib/.../SignalInterface"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Port"/>
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Resource"/>
    </InternalElement>
    <InternalLink Name="L1" RefPartnerSideA="GUID3:ConnectionPoint" RefPartnerSideB="GUID5:ConnectionPoint"/>
  </InternalElement>
</InstanceHierarchy>
```

Рисунок А.12 — Описание на языке XML понятия порта AutomationML Port




```
<SystemUnitClass Name="myPortClass">
  <Attribute Name="Direction">
    <Value>InOut</Value>
  </Attribute>
  <Attribute Name="Category">
    <Value>MaterialFlow</Value>
  </Attribute>
  <ExternalInterface Name="ConnectionPoint" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PortConnector"/>
  <SupportedRoleClass RefRoleClassPath="AutomationMLBaseRoleClassLib/.../Port"/>
</SystemUnitClass>
```

SystemUnitClass — Класс системных единиц; Name — Имя; Direction — Направление; Category — Категория; Value — Значение; InOut — Вход/Выход; MaterialFlow — Поток материала; ExternalInterface — Внешний интерфейс; RefBaseClassPath — Ссылочный путь доступа к базовому классу; SupportedRoleClass — Поддерживаемый ролевой класс; RefRoleClassPath — Ссылочный путь доступа к ролевому классу

Рисунок А.13 — Определение пользовательского класса портов AutomationML «myPortClass»

А.2.3 Понятие фасета AutomationML Facet

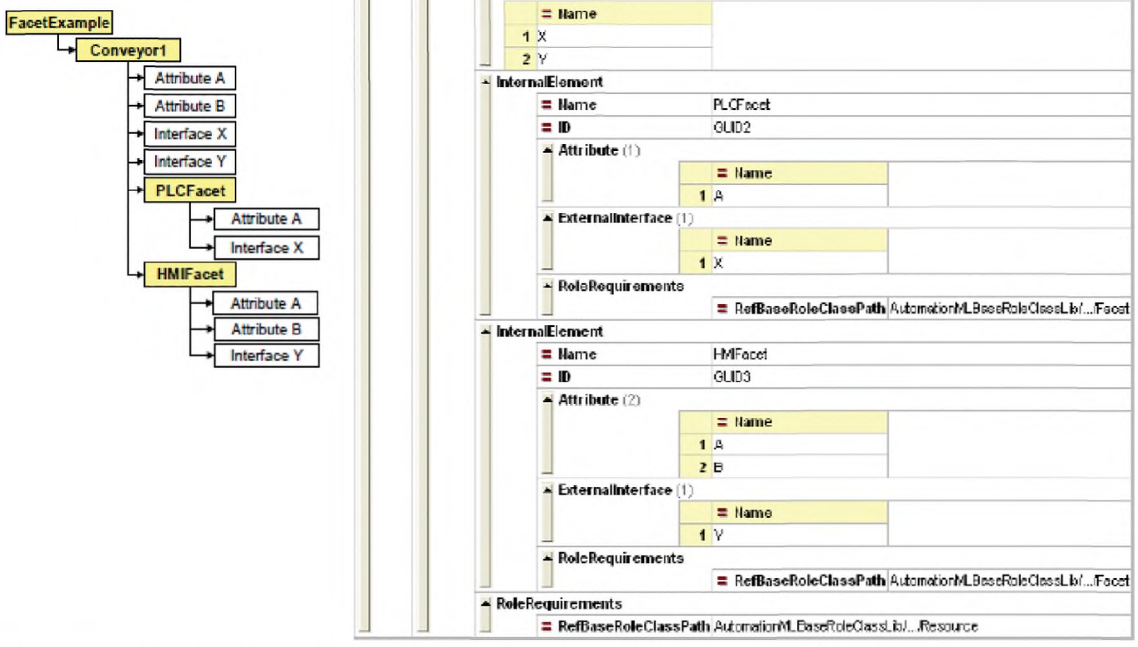
А.2.3.1 Описание понятия

Фасет — это объект языка AutomationML, обеспечивающий представление атрибутов и интерфейсов родительского объекта AutomationML. Данное понятие облегчает хранение различных настроек конфигурации (например, настроек человеко-машинного интерфейса HMI, программируемого логического PLC контроллера и т.д.) и способствует автоматизации некоторых шагов разработки системы управления производством. Кроме того, язык AutomationML определяет понятие ролевых классов AutomationML RoleClass «Facet» (см. раздел 6.4.4). Нормативные положения рассмотрены в разделе 8.3.

Описанная подгруппа атрибутов и интерфейсов относится к определенным инженерным аспектам. Она может хранить информацию о соответствующих инженерных решениях или шаблонах. Синтаксис и семантика указанных имен (значений) атрибутов в настоящем стандарте не рассматриваются. Они интерпретируются с помощью внешних средств инженерии, учитывающих синтаксис и семантику соответствующей информации. Таким образом, для работы указанных алгоритмов и выполнения задач автоматизированного проектирования нужна только информация о фасетах. Допустим, что атрибуты объекта включают имя шаблона кода программируемого логического PLC контроллера, а интерфейсы описывают входы/выходы данного шаблона. Получается, что алгоритм генерации кода PLC контроллера, учитывающий семантику рассматриваемых атрибутов и интерфейсов, может генерировать код PLC контроллера из имеющейся информации. То же самое происходит и с шаблоном человеко-машинного интерфейса HMI. Вышеупомянутые внешние алгоритмы, а также семантика соответствующих атрибутов (интерфейсов) в МЭК 62714 не рассматриваются. Выполнение задач автоматизированного проектирования осуществляется за счет совместного использования понятий AutomationML Group и AutomationML Facet.

А.2.3.2 Пример

Рисунок А.14 разъясняет понятие фасета AutomationML Facet на конкретном примере. Объект «Конвейер1» включает атрибуты «А» и «В», а также интерфейсы «X» и «Y». Назначенный объект фасета «PLCFacet» ссылается



InstanceHierarchy — Иерархия экземпляров; Name — Имя; InternalElement — Внутренний элемент; ID — Идентификатор; Attribute — Атрибут; ExternalInterface — Внешний интерфейс; RoleRequirements — Требования роли; RefBaseRoleClassPath — Ссылочный путь доступа к базовому ролевому классу; RoleRequirements — Требования роли; FacetExample — Пример фасета; Conveyor1 — Конвейер 1; Attribute A — Атрибут А; Interface X — Интерфейс X; PLCFacet — Фасет программируемого логического PLC контроллера; HMIFacet — Фасет человеко-машинного интерфейса HMI

Рисунок А.14 — Пример фасета AutomationML

```
<InstanceHierarchy Name="FacetExample">
  <InternalElement Name="Conveyor1" ID="GUID1">
    <Attribute Name="A"/>
    <Attribute Name="B"/>
    <ExternalInterface Name="X"/>
    <ExternalInterface Name="Y"/>
    <InternalElement Name="PLCFacet" ID="GUID2">
      <Attribute Name="A"/>
      <ExternalInterface Name="X"/>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Facet"/>
    </InternalElement>
    <InternalElement Name="HMIFacet" ID="GUID3">
      <Attribute Name="A"/>
      <Attribute Name="B"/>
      <ExternalInterface Name="Y"/>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Facet"/>
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Resource"/>
  </InternalElement>
</InstanceHierarchy>
```

Рисунок А.15 — Пример описания фасета AutomationML на языке XML

на атрибут «A» и интерфейс «X». При этом назначенный объект фасета «HMIFacet» ссылается на атрибуты «A» и «B» и интерфейс «Y». Поэтому оба фасета обеспечивают фильтрованное представление для имеющейся инженерной информации, необходимой для выполнения задач автоматизированного проектирования.

Пример применения: атрибут «A» может быть «ориентированным-на-продавца» именем специального шаблона кода программируемого логического PLC контроллера, описывающего функциональные возможности объекта «Конвейер1». Интерфейс «X» может быть именем входного сигнала, используемого данным шаблоном кода. Атрибут «B» может быть именем специального шаблона человеко-машинного интерфейса HMI конвейера. Интерфейс «Y» может быть сигналом, представляемым данным человеко-машинным интерфейсом HMI. С учетом данной информации, рассматриваемый программируемый логический контроллер PLC (человеко-машинный интерфейс HMI) способен генерировать инженерные решения автоматически. См. пример в разделе А.2.4.4.

А.2.4 Понятие группы AutomationML Group

А.2.4.1 Описание понятия

Понятие группы AutomationML Group позволяет отделить информацию о структуре от информации об экземпляре. Так как различные инженерные средства (на неоднородном множестве инструментальных средств инженерии) могут использовать различные представления на одни и те же данные, может оказаться полезным хранить данные представления раздельно. Это достигается за счет использования понятия AutomationML Group. Структурирование идентичных объектов возможно в различных иерархиях.

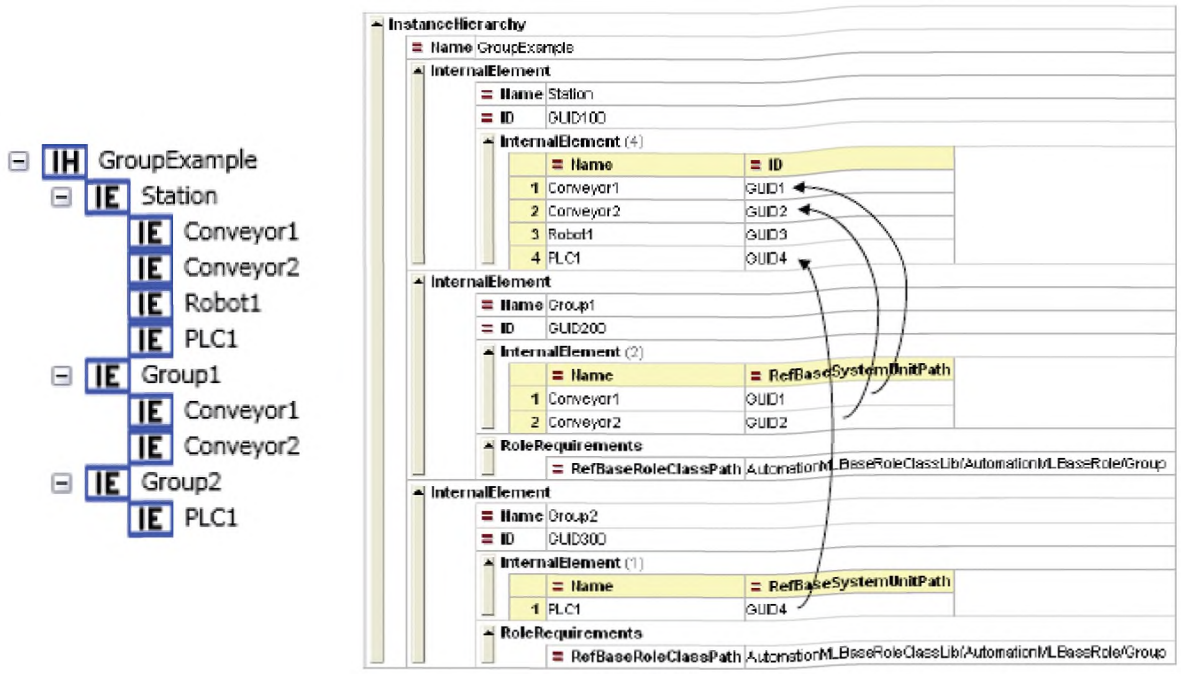
Путем определения атрибута группы «AssociatedFacet» (ассоциированный фасет), данная группа может быть ассоциирована с типом фасетов, характеризуемых уникальным именем. Это позволяет внешним инженерным алгоритмам автоматически идентифицировать рассматриваемые объекты (и их соответствующие фасеты), чтобы получить требуемую инженерную информацию. Кроме того, язык AutomationML определяет ролевой класс AutomationML RoleClass «Group» (см. раздел 6.4.3). Нормативные положения приведены в разделе 8.4.

А.2.4.2 Пример

Рисунок А.16a) описывает понятие группы с помощью структуры «Station (Станция)», содержащей объекты «Конвейер1», «Конвейер2», «Robot1» и «PLC1». Объекты «Group1» и «Group2» описывают одинаковые данные в различных иерархиях. Объект «Group1» определяет структурное представление только для конвейеров. Объект «Group2» отображает только объекты, относящиеся к программируемому логическому контроллеру PLC. В соответствии с МЭК 62424 (раздел А.2.14), формат CAEX обеспечивает хранение таких пересекающихся структур. Рисунок А.16b) содержит интерпретацию данного примера на языке AutomationML. Рисунок А.17 содержит соответствующее описание на языке XML.

А.2.4.3 Комбинация понятия группы и понятия фасета

Рисунок А.18 содержит пример возможной комбинации понятия группы и понятия фасета. Показанная иерархия экземпляров InstanceHierarchy отображает объект языка AutomationML «Station», включающий объекты AutomationML «Конвейер1» и «Конвейер2». Указанные конвейеры имеют два атрибута и два интерфейса каждый.



a) Hierarchy — a) Иерархия; b) Grid view — b) Сетевое представление; IH — Иерархия экземпляров; GroupExample — Пример группы; IE — Внутренний элемент; Station — Станция; Conveyor1 — Конвейер; PLC1 — Программируемый логический контроллер 1

Рисунок А.16 — Пример группы AutomationML Group

```
<InstanceHierarchy Name="GroupExample">
  <InternalElement Name="Station" ID="GUID100">
    <InternalElement Name="Conveyor1" ID="GUID1"/>
    <InternalElement Name="Conveyor2" ID="GUID2"/>
    <InternalElement Name="Robot1" ID="GUID3"/>
    <InternalElement Name="PLC1" ID="GUID4"/>
  </InternalElement>
  <InternalElement Name="Group1" ID="GUID200">
    <InternalElement Name="Conveyor1" RefBaseSystemUnitPath="GUID1"/>
    <InternalElement Name="Conveyor2" RefBaseSystemUnitPath="GUID2"/>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group"/>
  </InternalElement>
  <InternalElement Name="Group2" ID="GUID300">
    <InternalElement Name="PLC1" RefBaseSystemUnitPath="GUID4"/>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group"/>
  </InternalElement>
</InstanceHierarchy>
```

Рисунок А.17 — Пример описания группы AutomationML Group на языке XML

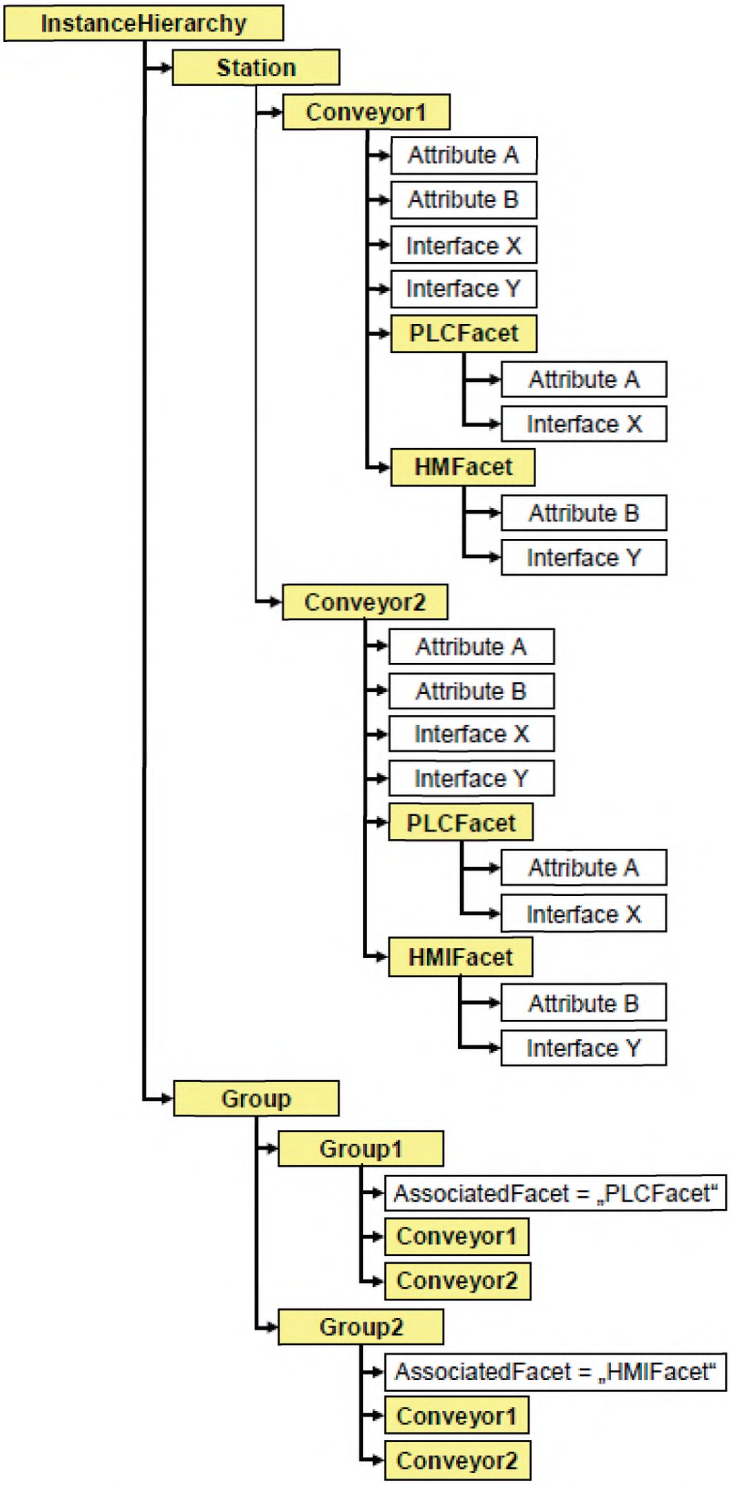
Объект языка AutomationML «Group» представляет собой встроенные группы «Group1» и «Group2». Обе группы ссылаются на объекты конвейера, но имеют различные ассоциации фасетов.

Пример применения: Алгоритм генерации кода управления может пройти по иерархии экземпляров InstanceHierarchy и идентифицировать все группы, ассоциированные с фасетом «PLCFacet», а затем сгенерировать код для оценки ссылочных объектов.

Рисунок А.19 содержит описание на языке XML, соответствующее примеру, приведенному на рисунке А.18.

А.2.4.4 Автоматическая генерация человеко-машинного интерфейса HMI с помощью понятия группы и понятия фасета

В данном примере принято, что атрибут «B» конвейера представляет собой шаблон интерфейса HMI, визуализирующий переменную «Y». Рисунок А.20 иллюстрирует характерный шаблон интерфейса HMI конвейера.



InstanceHierarchy — Иерархия экземпляров; Station — Станция; Conveyor1 — Конвейер 1; Attribute A — Атрибут A; Interface X — Интерфейс X; PLCFacet — Фасет программируемого логического контроллера; HMFacet — Фасет человеко-машинного интерфейса; Group — Группа; AssociatedFacet — Ассоциированный фасет

Рисунок А.18 — Комбинация понятия группы и понятия фасета

```
<InstanceHierarchy Name="FacetGroupCombination">
  <InternalElement Name="Conveyor1" ID="GUID1">
    <Attribute Name="A"/>
    <Attribute Name="B"/>
    <ExternalInterface Name="X"/>
    <ExternalInterface Name="Y"/>
    <InternalElement Name="PLCFacet" ID="GUID2">
      <Attribute Name="A"/>
      <ExternalInterface Name="X"/>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Facet"/>
    </InternalElement>
    <InternalElement Name="HMIFacet" ID="GUID3">
      <Attribute Name="B"/>
      <ExternalInterface Name="Y"/>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Facet"/>
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource"/>
  </InternalElement>
  <InternalElement Name="Conveyor2" ID="GUID4">
    <Attribute Name="A"/>
    <Attribute Name="B"/>
    <ExternalInterface Name="X"/>
    <ExternalInterface Name="Y"/>
    <InternalElement Name="PLCFacet" ID="GUID5">
      <Attribute Name="A"/>
      <ExternalInterface Name="X"/>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Facet"/>
    </InternalElement>
    <InternalElement Name="HMIFacet" ID="GUID6">
      <Attribute Name="B"/>
      <ExternalInterface Name="Y"/>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Facet"/>
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Resource"/>
  </InternalElement>
  <InternalElement Name="Group" ID="GUID7">
    <InternalElement Name="Group1" ID="GUID8">
      <Attribute Name="AccociatedFacet">
        <Value>PLCFacet</Value>
      </Attribute>
      <InternalElement Name="Conveyor1" RefBaseSystemUnitPath="GUID1"/>
      <InternalElement Name="Conveyor2" RefBaseSystemUnitPath="GUID2"/>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group"/>
    </InternalElement>
    <InternalElement Name="Group2" ID="GUID9">
      <Attribute Name="AccociatedFacet">
        <Value>HMIFacet</Value>
      </Attribute>
      <InternalElement Name="Conveyor1" RefBaseSystemUnitPath="GUID1"/>
      <InternalElement Name="Conveyor2" RefBaseSystemUnitPath="GUID2"/>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group"/>
    </InternalElement>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/Group"/>
  </InternalElement>
</InstanceHierarchy>
```

Рисунок А.19 — Описание представления комбинации «Фасет-Группа» на языке XML

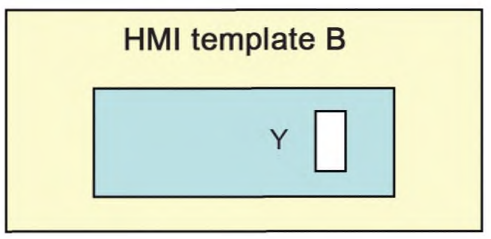


Рисунок А.20 — Характерный шаблон В для интерфейса HMI, визуализирующий переменную «Y» технологического процесса конвейера

Основанный на конкретном экземпляре конвейера, рассматриваемый инженерный алгоритм идентифицирует, что объект языка AutomationML «Group2» ассоциирован с фасетом человеко-машинного интерфейса HMI. Здесь он идентифицирует, что экземпляры «Конвейер1» и «Конвейер2» являются частью интерфейса HMI. Данный алгоритм может извлекать информацию из интерфейса HMI о каждом из двух конвейеров. Он может идентифицировать соответствующий шаблон интерфейса HMI, может ассоциировать корректные сигналы для последующей их визуализации. Рисунок А.21 представляет результирующий человеко-машинный интерфейс HMI.

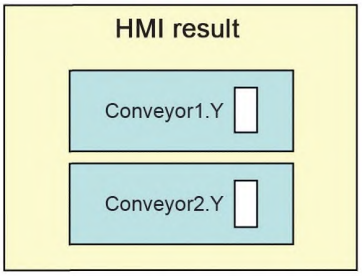


Рисунок А.21 — Сгенерированный результирующий интерфейс HMI «B», визуализирующий оба конвейера с индивидуальными переменными технологического процесса

А.2.5 Понятие набора свойств PropertySet

А.2.5.1 Описание понятия

Набор свойств PropertySet применяется как таксономия (структурированный словарь) атрибутов. Он моделируется как ролевой класс, содержащий предварительно определенные атрибуты, описывающие свойства рассматриваемой области применения и получен из стандартного ролевых классов «PropertySet» (см. раздел 6.4.13). Данный набор включает список синтаксически и семантически согласованных атрибутов. Наборы свойств собраны в библиотеки ролевых классов.

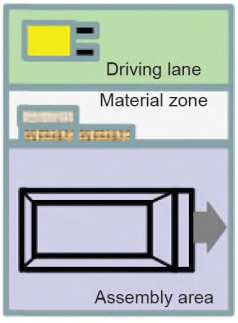
Объекты AutomationML могут быть ассоциированы с одним или несколькими наборами свойств. Для каждого набора свойств создается отдельный дочерний объект типа CAEX InternalElements, который назначается соответствующему классу PropertySet с помощью определений требования роли RoleRequirement. Несколько наборов свойств могут быть ассоциированы с помощью нескольких дочерних внутренних элементов InternalElements соответствующего объекта AutomationML.

Объект отображения CAEX обеспечивает отображение собственных атрибутов объекта AutomationML с предварительно семантически определенными атрибутами набора свойств PropertySet. Это обеспечивает возможность импортеру программного обеспечения автоматически интерпретировать указанные атрибуты и отображать их на целевые специальные атрибуты инструментальных средств. Данная процедура упрощает автоматический обмен данных между различными инструментами.

Также существуют предварительно определенные библиотеки наборов свойств AutomationML. Нормативные положения приведены в разделе 8.5.

А.2.5.2 Пример

В качестве примера (на языке AutomationML) рассматривается процедура передачи схемы линии сборки, содержащей станции сборки (см. рисунок А.22). Станция сборки состоит их трех площадок. Одна — для транспортировки материала, вторая — для хранения материала, третья — для непосредственной сборки (см. ниже). Средства инженерии источника определяют данные площадки с помощью пользовательских атрибутов, назначенных объекту и представляющих рассматриваемую станцию.



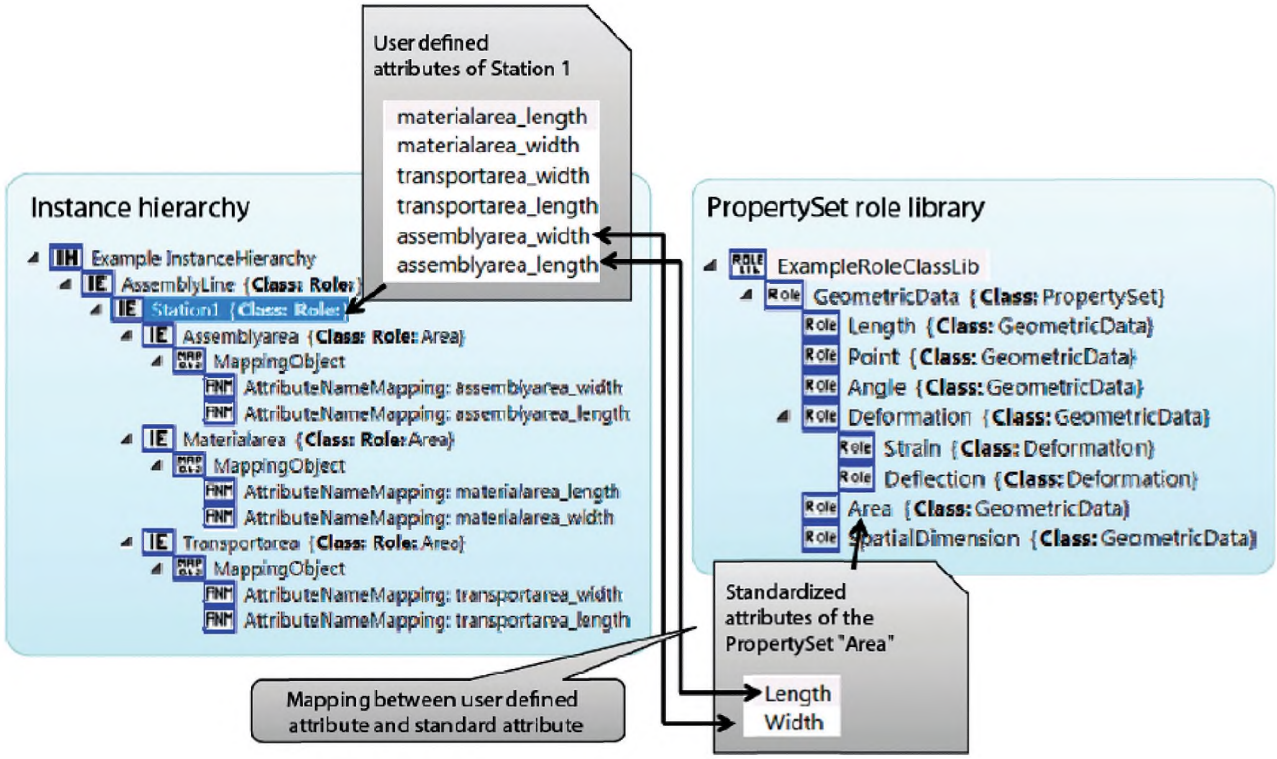
Driving lane — Полоса движения; Material zone — Зона хранения материала; Assembly area — Сборочная площадка

Рисунок А.22 — Пример набора свойств PropertySet

Рисунок А.23 иллюстрирует соответствующую модель AutomationML. В левой части рисунка показана иерархия экземпляров, необходимая для моделирования трех площадок Станции 1. Каждая площадка имеет свой пользовательский набор атрибутов и ссылку на набор свойств PropertySet «Area». Соответствующее описание на языке XML приведено на рисунке А.24.

В правой части рисунка А.23 приведена библиотека набора свойств AutomationML PropertySet. Данная библиотека ролевых классов AutomationML RoleClass содержит класс RoleClass «GeometricData» (геометрические данные), полученный из базового ролевых классов Base-RoleClass «PropertySet». Ролевой класс RoleClass «GeometricData» сам имеет дополнительные производные, определяющие некоторые базовые геометрические свойства. Класс RoleClass с именем «Area» (площадка) определяет атрибуты «Length» (длина) и «Width» (ширина) как атрибуты типа «double» (двойной) и единицу измерения «m» (метр). На рисунке А.25 приведено описание на языке XML, определяющее рассматриваемые атрибуты набора свойств PropertySet.

С помощью набора свойств PropertySets экспортирующий инструмент может сохранять пользовательские объекты с пользовательскими атрибутами и объяснять семантику пользовательских атрибутов с помощью процедуры отображения. Данное утверждение поддерживает автоматическую интерпретацию рассматриваемых атрибутов. Таким образом, целевой инженерный инструмент может интерпретировать данные и выполнять преобразование единиц измерения, используемых атрибутами набора свойств PropertySet.



User defined attributes of Station 1 — Пользовательские атрибуты станции 1; materialarea_length — Длина площадки хранения материала; materialarea_width — Ширина площадки хранения материала; transportarea_width — Ширина площадки транспортировки; transportarea_length — Длина площадки транспортировки; assemblyarea_width — Ширина сборной площадки; assemblyarea_ length — Длина сборной площадки; Instance hierarchy — Иерархия экземпляров; PropertySet role library — Библиотека ролей набора свойств PropertySet; Mapping between user defined attribute and standard attribute — Отображения между пользовательскими атрибутами и стандартными атрибутами; Standardized attributes of the PropertySet «Area» — Стандартные атрибуты набора данных PropertySet «Area»; Length — Длина; Width — Ширина; Example InstanceHierarchy — Пример иерархии экземпляров; AssemblyLine (Class: Role) — Линия сборки (класс: роль); Station 1(Class Role) — Станция 1 (класс: роль:); Assemblyarea (Class: Role: Area) — Площадка сборки (класс роль: площадка); MappingObject — Объект отображения; AttributeNameMapping: assemblyarea_width — Отображение имени атрибута: ширина сборной площадки; Materialarea (Class: Role: Area) — площадка хранения материала (класс: роль: площадка); Transportarea (Class: Role: Area) — Площадка транспортировки (класс: роль: площадка); ExampleRoleClassLib — Пример библиотеки ролевых классов; GeomatricData (Class: PropertySet) — Геометрические данные (класс: набор свойств); Length (Class: GeometricData) — Длина (класс: геометрические данные); Point — Точка; Angle — Угол; Deformation — Деформация; Strain — Напряжение; Defection — Отклонение; Area — Площадь; SpatialDimension — Пространственный размер; Role — Роль

Рисунок А.23 — Пример набора свойств PropertySet

```
<InstanceHierarchy Name="Example InstanceHierarchy">
  <InternalElement Name="AssemblyLine" ID="{8b2510b4-7fc5-4f54-9d09-a3197a062604}">
    <InternalElement Name="Station1" ID="{54500138-c6a1-47c8-80c9-bee35662563c}">
      <Attribute Name="materialarea_length" />
      <Attribute Name="materialarea_width" />
      <Attribute Name="transportarea_width" />
      <Attribute Name="transportarea_length" AttributeDataType="xs:double" />
      <Attribute Name="assemblyarea_width" AttributeDataType="xs:double" />
      <Attribute Name="assemblyarea_length" AttributeDataType="xs:double" />
      <InternalElement Name="Assemblyarea" ID="{e08f53e5-eb0f-45c8-b42f-4714824dd05c}">
        <RoleRequirements RefBaseRoleClassPath="ExampleRoleClassLib/GeometricData/Area" />
        <MappingObject>
          <AttributeNameMapping SystemUnitAttributeName="assemblyarea_width" RoleAttributeName="Width" />
          <AttributeNameMapping SystemUnitAttributeName="assemblyarea_length" RoleAttributeName="Length" />
        </MappingObject>
      </InternalElement>
      <InternalElement Name="Materialarea" ID="{668360af-d108-4b60-be53-ddb262bc470e}">
        <RoleRequirements RefBaseRoleClassPath="ExampleRoleClassLib/GeometricData/Area" />
        <MappingObject>
          <AttributeNameMapping SystemUnitAttributeName="materialarea_length" RoleAttributeName="Length" />
          <AttributeNameMapping SystemUnitAttributeName="materialarea_width" RoleAttributeName="Width" />
        </MappingObject>
      </InternalElement>
      <InternalElement Name="Transportarea" ID="{19418967-cd7c-46ea-adea-81aa52f6b685}">
        <RoleRequirements RefBaseRoleClassPath="ExampleRoleClassLib/GeometricData/Area" />
        <MappingObject>
          <AttributeNameMapping SystemUnitAttributeName="transportarea_width" RoleAttributeName="Width" />
          <AttributeNameMapping SystemUnitAttributeName="transportarea_length" RoleAttributeName="Length" />
        </MappingObject>
      </InternalElement>
    </InternalElement>
  </InternalElement>
</InstanceHierarchy>
```

Рисунок А.24 — Описание на языке XML иерархии экземпляров

А.2.6 Понятие «Процесс-Продукт-Ресурс»

А.2.6.1 Описание понятия

В процессе накопления практического опыта структурирования комплексных инженерных данных технологической установки, установлено, что необходима трисекция данных на ресурсы, процессы и продукты. Рассматриваемое понятие используется в различных областях, например, для цифровых производственных инструментальных средств (цифровая модель производства) в соответствии с МЭК 62264 на уровне системы организации производства (MES).

- Ресурсы — это центральный компонент рассматриваемой модели. Они обеспечивают производство и обработку продукта. В языке AutomationML ресурс — это сущность, используемая для производства, включая установки, роботов, станки, их состояние, оборудование, возможные сообщения и т. д. Ресурсы могут быть как аппаратными компонентами производственной системы, так и программным обеспечением данной системы (например, система SCADA). В языке AutomationML ресурсы обычно моделируются в иерархии производственной установки, образуя ее топологию.

- Изготовленный продукт — основная цель рассмотрения. Он определяет, какие технологические процессы следует применить к материалам (промежуточным продуктам), какое оборудование следует использовать. Продукт рассматривается в рамках непрерывного, дискретного или серийного производства. Продукт на языке AutomationML отображает готовый товар. Он может формироваться иерархически. Рассматриваемый продукт не обязательно должен быть конечным. К продукту относят также результаты его испытаний, спецификации и сопутствующую документацию.

- Технологический процесс также является центральным элементом модели. Процесс на языке AutomationML представляет собой производственный процесс, включающий несколько подпроцессов. Параметры процесса, технологические цепочки, планирование процесса — неотъемлемые его части. Процессы модифицируют продукты. Язык AutomationML отражает получение конечного продукта из различных подпродуктов, отражает изменение веществ путем их химической обработки. Процессы связаны с ресурсами и наоборот.

В каждом случае, представления ресурсов, продуктов и процессов связаны друг с другом (см. рисунок А.26). Корректным назначением для процесса «transport» является ресурс «conveyor» (конвейер). Ресурс «press» (прессование) обеспечивает «scrap cubes» (спрессованные отходы). Процесс «welding» (сварка) обеспечивает «weld» (сварку) двух «metal» (металлических заготовок).

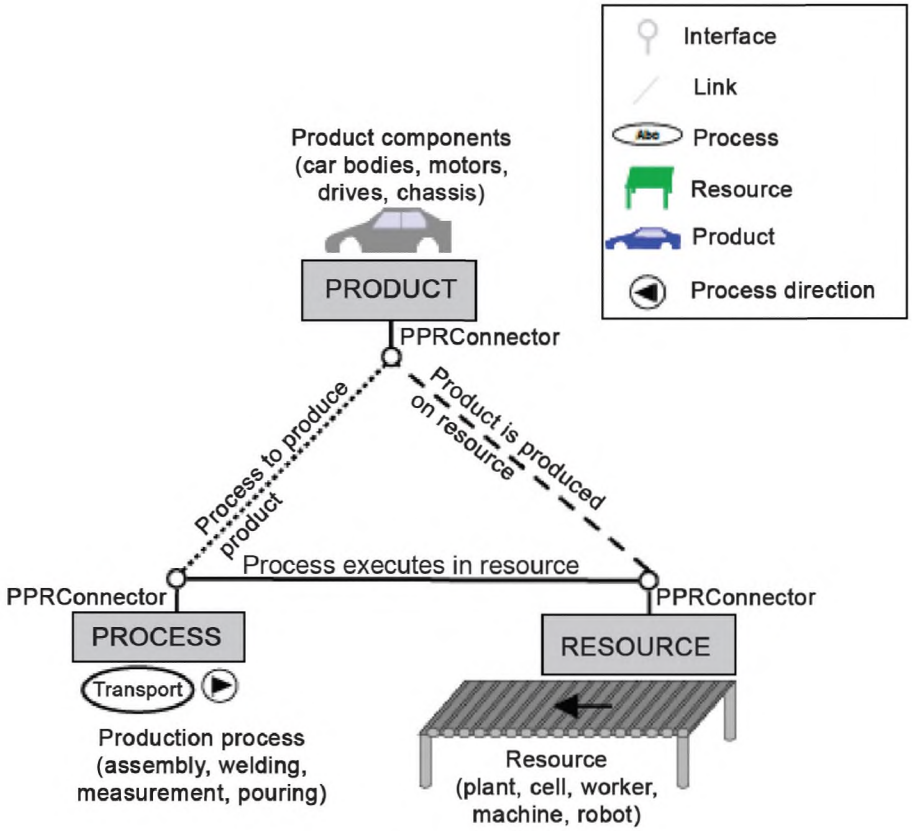
```
- <RoleClass Name="GeometricData"
    RefBaseClassPath="AutomationMLBaseRoleClassLib/AutomationMLBaseRole/PropertySet
  - <RoleClass Name="Length"
      RefBaseClassPath="ExampleRoleClassLib/GeometricData">
      <Attribute Name="lengthValue" AttributeDataType="xs:double"
        Unit="m" />
    </RoleClass>
  - <RoleClass Name="Point"
      RefBaseClassPath="ExampleRoleClassLib/GeometricData">
      <Attribute Name="xAxis" AttributeDataType="xs:double" />
      <Attribute Name="yAxis" AttributeDataType="xs:double" />
      <Attribute Name="zAxis" AttributeDataType="xs:double" />
    </RoleClass>
  - <RoleClass Name="Angle"
      RefBaseClassPath="ExampleRoleClassLib/GeometricData">
      <Attribute Name="angleValue" AttributeDataType="xs:double"
        Unit="rad" />
    </RoleClass>
  - <RoleClass Name="Deformation"
      RefBaseClassPath="ExampleRoleClassLib/GeometricData">
      <Description>Deformation of the material is the change in geometry
        when stress is applied (in the form of force loading, gravitational
        field, acceleration, thermal expansion, etc.). Deformation is
        expressed by the displacement field of the material</Description>
      <Attribute Name="deformationValue" AttributeDataType="xs:double" />
    - <RoleClass Name="Strain"
        RefBaseClassPath="ExampleRoleClassLib/GeometricData/Deformation">
        <Description>Strain is the deformation per unit length</Description>
      </RoleClass>
    - <RoleClass Name="Deflection"
        RefBaseClassPath="ExampleRoleClassLib/GeometricData/Deformation">
        <Description>Deflection is a term to describe the magnitude to
          which a structural element bends under a load</Description>
      </RoleClass>
    </RoleClass>
  - <RoleClass Name="Area"
      RefBaseClassPath="ExampleRoleClassLib/GeometricData">
      <Attribute Name="Length" AttributeDataType="xs:double" Unit="m" />
      <Attribute Name="Width" AttributeDataType="xs:double" Unit="m" />
    </RoleClass>
  - <RoleClass Name="SpatialDimension"
      RefBaseClassPath="ExampleRoleClassLib/GeometricData">
      <Attribute Name="XAxis" AttributeDataType="xs:double" Unit="m" />
      <Attribute Name="YAxis" AttributeDataType="xs:double" Unit="m" />
      <Attribute Name="ZAxis" AttributeDataType="xs:double" Unit="m" />
    </RoleClass>
  </RoleClass>
```

Рисунок А.25 — Пример библиотеки AutomationML PropertySet в кодах на языке XML

Для создания связей между указанными элементами требуется наличие интерфейса. Для этого язык AutomationML определяет стандартный класс интерфейсов PPRConnector (см. рисунок А.27). Нормативные положения в части коннектора PPRConnector рассмотрены в разделе 6.3.5. С помощью данного интерфейса могут быть установлены связи между элементами с помощью стандартных внутренних связей CAEX InternalLinks (см. раздел 5.6.6). Таким образом, ресурсы могут быть связаны с продуктами, которыми можно манипулировать.

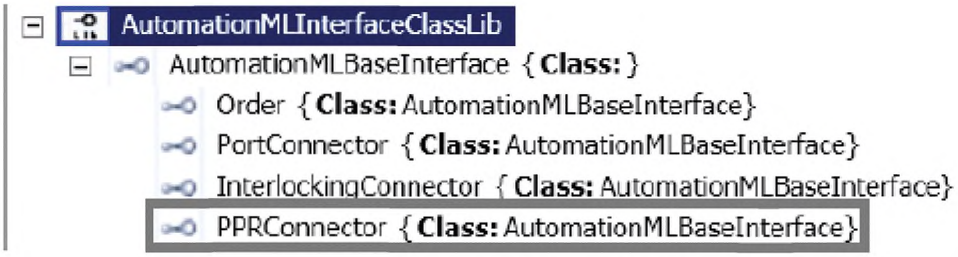
А.2.6.2 Пример

Следующий пример (см. рисунок А.28) иллюстрирует использование данного понятия в языке AutomationML. Оно включает два конвейера (С1 и С2), поворотную платформу (ТТ1) и робота (RB1). Это ресурсы производственной установки. Робот устанавливает колеса на автомобиль. Колеса и автомобиль — продукты. Производственные процессы в рамках рассматриваемо примера: транспортирование, поворот платформы, сборка.



Interface — Интерфейс; Link — Связующее звено; Process — Процесс; Resource — Ресурс; Product — Продукт; Process direction — Направление процесса; Product components (car bodies, motors, drives, chassis) — Компоненты продукта (автомобиля): кузов, двигатель, привод, шасси; PPRConnector — Коннектор PPRConnector; Process to produce product — Процесс для производства продукта; Product is produced on resource — Продукт производится с помощью ресурса; Transport — Транспорт; Production process (assembly, welding, measurement, pouring) — Производственный процесс (сборка, сварка, измерение, заливка); Resource (plant, cell, worker, machine, robot) — Ресурс (промышленная установка, производственная ячейка, рабочий, станок, робот)

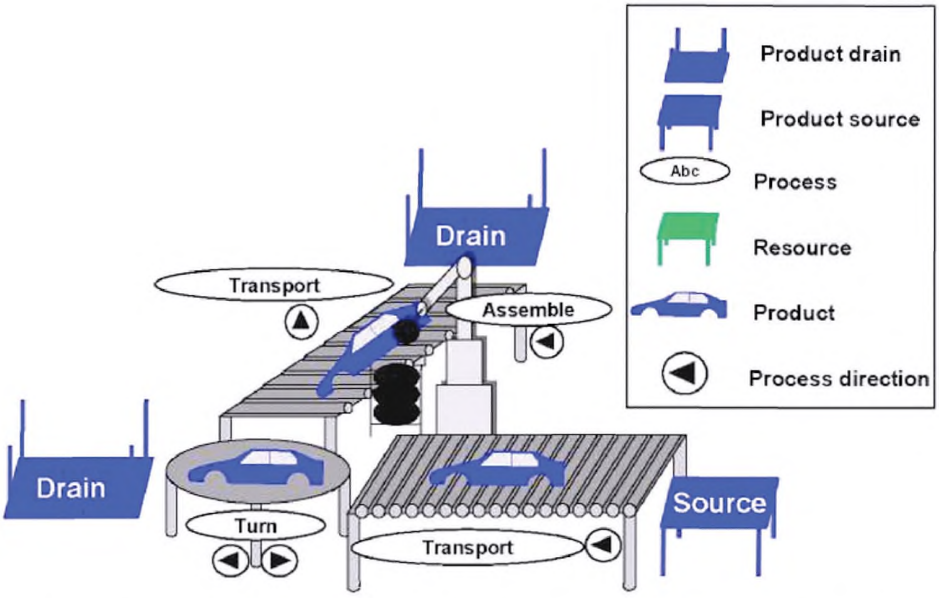
Рисунок А.26 — Базовые элементы понятия «Продукт—Процесс—Ресурс»



AutomationMLInterfaceClassLib — Библиотека класса интерфейсов; AutomationMLBaseInterface (class:) — Базовый интерфейс (класс:); Order — Порядок; PortConnector — Коннектор порта; InterlockingConnector — Взаимосвязанные коннекторы; PPRConnector — Коннектор PPRConnector

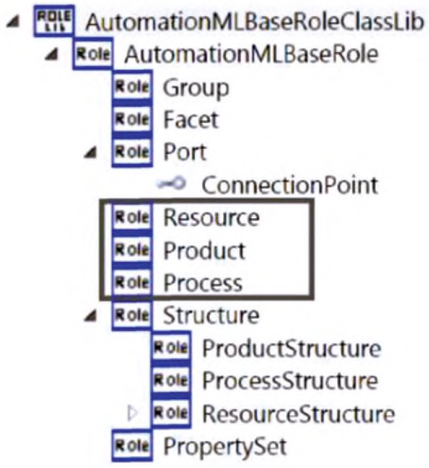
Рисунок А.27 — Интерфейс коннектора PPRConnector

На языке AutomationML понятие «Процесс—Продукт—Ресурс» моделируется с помощью понятия роли CAEX (см. МЭК 62424 (раздел А.2.9)) и соотношений между элементами (см. раздел 5.6.6). Наборы элементов с назначенными ролями «Ресурс», «Продукт» или «Процесс» разбиваются на пары. Это означает, что ресурс не может быть и продуктом, и процессом одновременно. Соответствующие классы ролей являются частью библиотеки базового ролевых классов AutomationMLBaseRoleClassLib (см. рисунок А.29 и раздел 6.4).



Product drain — Выпуск продукта; Product source — Источник продукта; Process — Процесс; Resource — Ресурс; Product — Продукт; Process direction — Направление процесса; Drain — Выпуск; Transport — Транспортирование; Assemble — Сборка; Turn — Поворот платформы

Рисунок А.28 — Пример понятия «Продукт—Процесс—Ресурс»

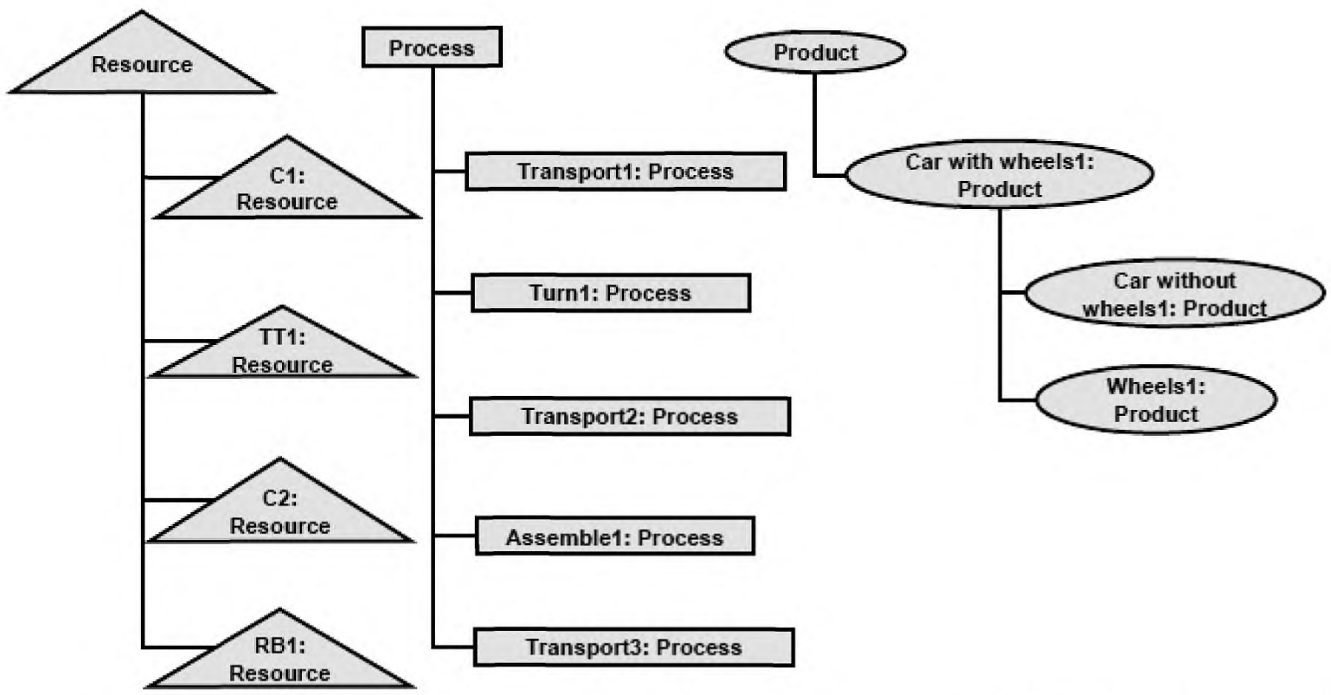


AutomationMLBaseRoleClassLib — Библиотека базового ролевых классов; AutomationMLBaseRole — Базовая роль; Group — Группа; Facet — Фасет; Port — Порт; Role Lib — Библиотека ролей; Role — Роль; ConnectionPoint — Точка соединения; Resource — Ресурс; Product — Продукт; Process — Процесс; Structure — Структура; PropertySet — Набор свойств

Рисунок А.29 — Роли AutomationML, необходимые для понятия «Процесс—Продукт—Ресурс»

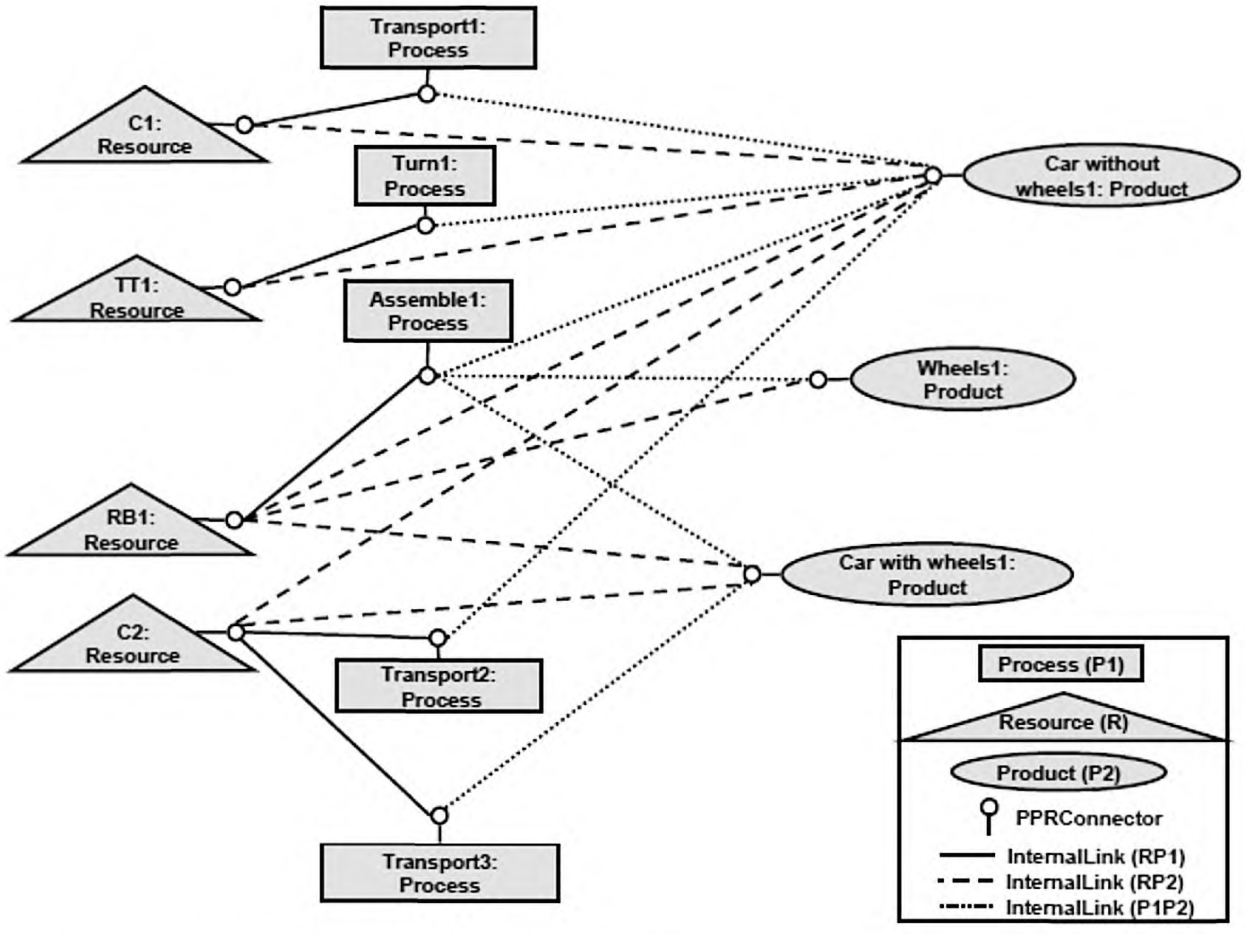
В рассматриваемом примере (см. рисунок А.28) роль «Ресурс» назначена конвейеру, роботу и поворотной платформе. Автомобилям и колесам назначена роль «Продукт». Роль «Процесс» соответствует процессам транспортировки, поворота платформы и сборки. Все элементы хранятся на соответствующем поддереве (см. рисунок А.30). Порядок применения процессов, продуктов и ресурсов явно выражается связующими элементами, установленными между соответствующими интерфейсами типа «Order» (это не показано на рисунке, чтобы его не загромождать).

Каждый элемент (в рассматриваемом примере) имеет интерфейс коннектора PPRConnector. Полные представления связующих элементов представлены на рисунке А.31. Непрерывные линии — это связи ресурсов с процессами. Точечные линии — это связи процессов с продуктами. Штриховые линии — это связи ресурсов с продуктами. Схема представляется достаточно сложной, поэтому (для простоты) избыточные связи опущены.



Resource — Ресурс; Process — Процесс; Product — Продукт; Transport1: Process — Транспортирование 1: процесс; Car with wheels1: Product — Автомобиль с колесами 1: продукт; Turn1: Process — Поворот платформы 1: процесс; Car without wheels1: Product — Автомобиль без колес 1: продукт; Wheels1: Product — Колеса 1: продукт; Assemble1: Process — Сборка 1: процесс

Рисунок А.30 — Элементы рассматриваемого примера



Transport1: Process — Транспорт 1: процесс; C1: Resource — C1: ресурс; Turn1: Process — Поворот платформы 1: процесс; Car without wheels1: Product — Автомобиль без колес 1: продукт; Assemble1: Process — Сборка 1: процесс; Wheels1: Product — Колеса 1: продукт; Car with wheels1: Product — Автомобиль с колесами 1: продукт; PPRConnector — Коннектор; InternalLinks — Внутренняя связь

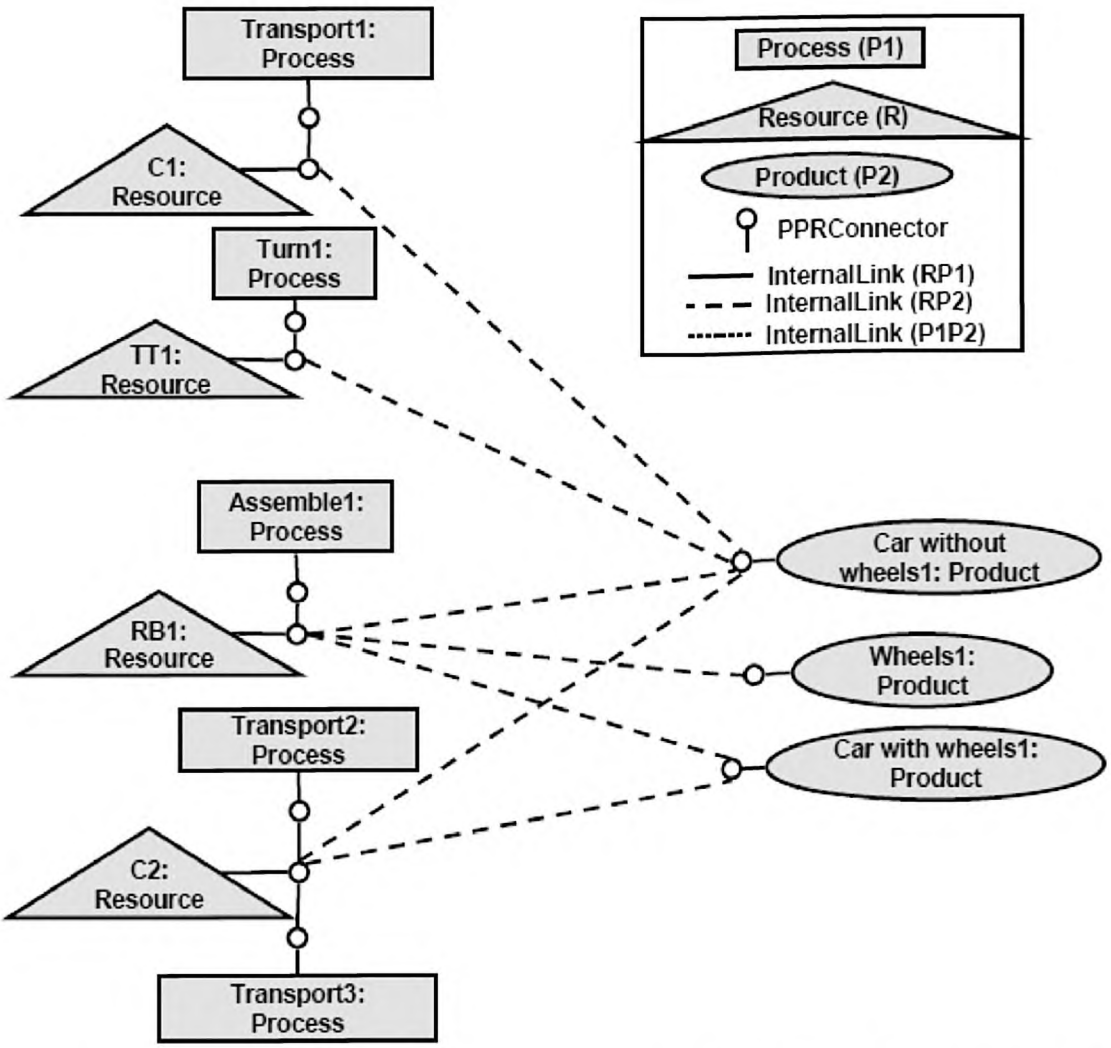
Рисунок А.31 — Связи, рассматриваемые в примере

На рисунке А.32 данный пример рассмотрен с точки зрения ресурса. Поэтому нужны только 12 связей из их общего количества, равного 19. Конвейер «С1» соединен с продуктом «Car without wheels1» (автомобиль без колес1) (см. также поворотную платформу «ТТ1» и конвейер «С2»). Также как робот устанавливает колеса на автомобиле на конвейере «С2», робот «RB1» соединен с продуктами «Car without wheels1» (автомобиль без колес1), «Car with wheels1» (автомобиль с колесами1) и «Wheels1» (колеса1). Дополнительно конвейер «С2» связан с продуктом «Car with wheels1» (автомобиль с колесами1). Процесс транспортировки «Transport1» связан с конвейером «С1». Процессы транспортировки «Transport2» и «Transport3» связаны с конвейером «С2». Процесс «Assemble1» связан с роботом «RB1». Процесс «Turn1» связан с поворотной платформой «ТТ1». Связи продуктов с процессами (точечные линии на рисунке А.31) могут быть получены из других существующих связей. Данную блок-схему можно произвольно поворачивать, продукты и процессы блок-схемы можно произвольно перемещать.

На рисунке А.33 приведено древо рассматриваемого объекта AutomationML. Внутренняя связь InternalLink между конвейером «С1» и процессом «Transport1» вынесена отдельно.

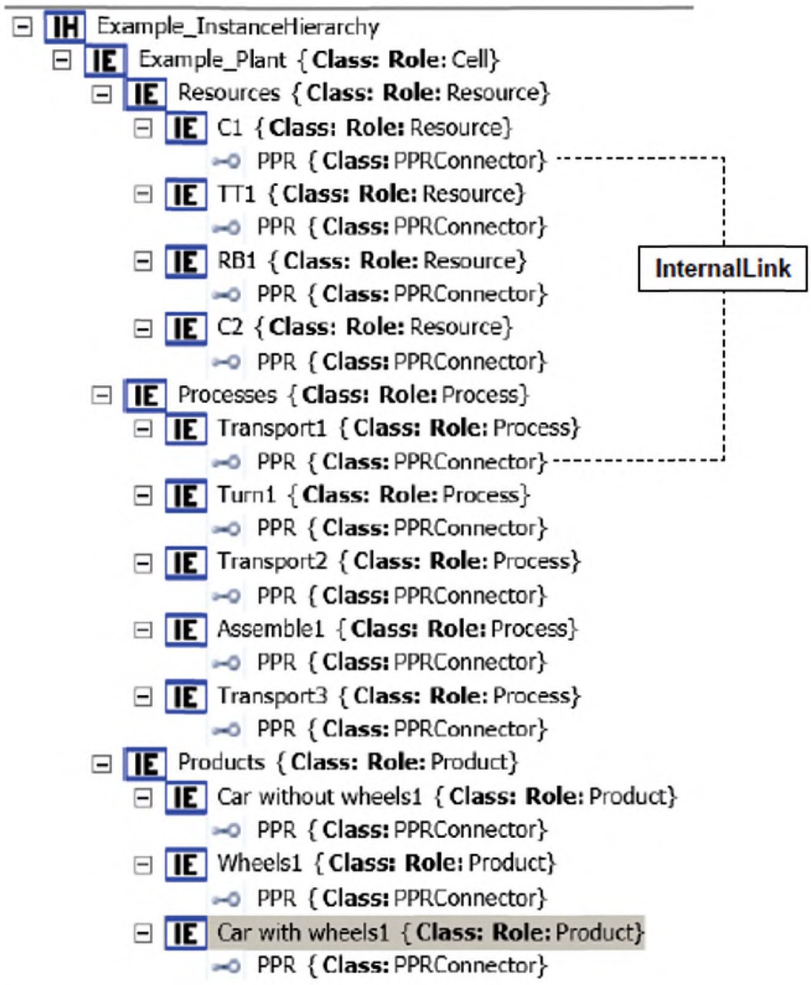
Соответствующая модель на языке XML приведена на рисунке А.34. На первом уровне примера приведены три базовых элемента («Ресурсы», «Процессы», «Продукты»), моделируемые как внутренние элементы в формате CAEX InternalElements.

Ниже объекта «Resource» расположены четыре компонента: два конвейера, поворотная платформа и робот. Они также имеют тип «InternalElements» (внутренний элемент). Данные компоненты имеют наружный интерфейс



Transport1: Process — Транспорт 1: процесс; C1: Resource — С1: ресурс; Turn1: Process — Поворот платформы 1: процесс; Car without wheels1: Product — Автомобиль без колес 1: продукт; Assemble1: Process — Сборка 1: процесс; Wheels1: Product — Колеса 1: продукт; Car with wheels1: Product — Автомобиль с колесами 1: продукт; PPRConnector — Коннектор PPRConnector; InternalLinks — Внутренняя связь

Рисунок А.32 — Рассмотрение примера с точки зрения ресурса



Example_InstanceHierarchy — Иерархия экземпляров в примере; Example_Plant (Class: Role: Cell) — Пример производственной установки (класс: роль: производственная ячейка); IH — Иерархия экземпляров; IE — Внутренний элемент; InternalLink — Внутренняя связь

Рисунок А.33 — Пример построения иерархии экземпляров InstanceHierarchy на языке AutomationML

(коннектор) ExternalInterface PPRConnector. Они назначают ролевой класс «Resource». Процессы и продукты имеют интерфейс, их роли распределены. Для организации связей между элементами, объекты InternalLinks обычно располагают на одном уровне как самые главные базовые элементы. Рассматриваемые связи показаны на языке XML на рисунке А.35.

Полный обзор понятий в рамках рассматриваемого примера приведен на рисунке А.36.

А.2.7 Поддержка сразу нескольких ролей

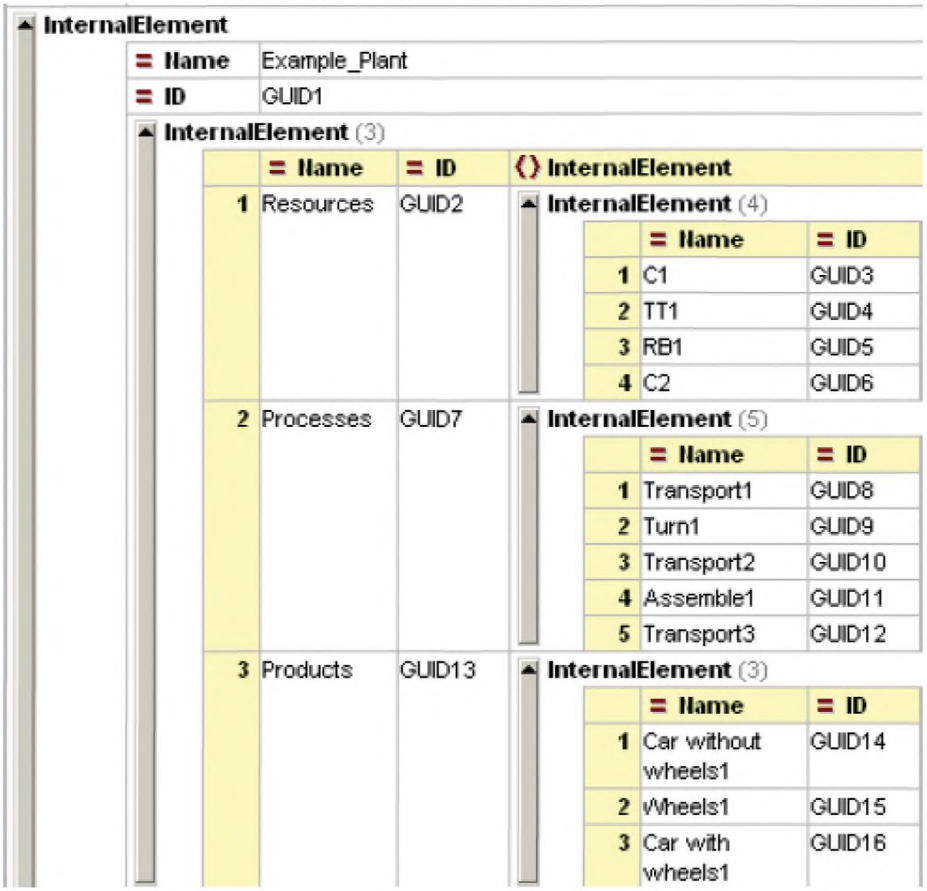
В добавление к МЭК 62424:2008 (раздел А.2.9) язык AutomationML обеспечивает возможность поддержки сразу нескольких ролей для экземпляра одного объекта. Рассмотрение сразу нескольких ролей имеет смысл, если данный объект имеет несколько функциональных возможностей.

Примером является многофункциональное устройство, работающее как сканер, принтер и факс одновременно. Положения, касающиеся одновременной поддержки нескольких ролей, рассмотрены в разделе 8.5.

Рисунок А.37 содержит пример, в котором объект «MultiDevice01» имеет три атрибута «Fax-BoudRate», «PrintSpeed (скорость печати)», «FaxSpeed» (скорость передачи сообщения) и два интерфейса «PowerSupply» (питание) и «USB» (USB-интерфейс)». Объект «MultiDevice01» поддерживает три роли «Printer» (принтер), «Fax» (факс), «Scanner» (сканер). Роль со ссылочным тэгом «RefBaseRoleClassPath» (ссылочный путь доступа к базовому ролевому классу) для соответствующего элемента требований RoleRequirement является базовой.

На рисунке А.38 приведен соответствующий код на языке.

Атрибуты и интерфейсы, принадлежащие объекту «MultiDevice01», должны отображаться на атрибуты и интерфейсы всех трех ассоциированных ролей. Это производится с помощью объекта отображения CAEX



InternalElement — Внутренний элемент; Name — Имя; ID — Идентификатор; Example_Plant — Производственная установка примера

Рисунок А.34 — Пример формирования внутренних элементов InternalElements

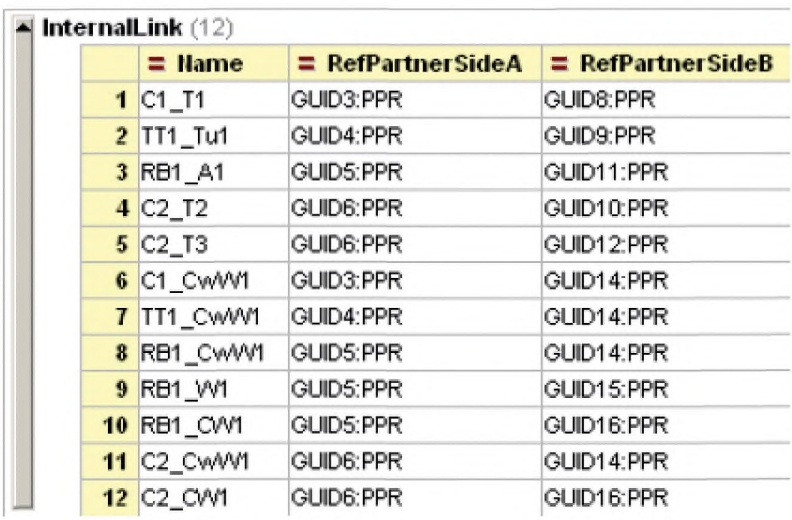


Рисунок А.35 — Пример организации внутренних связей InternalLinks

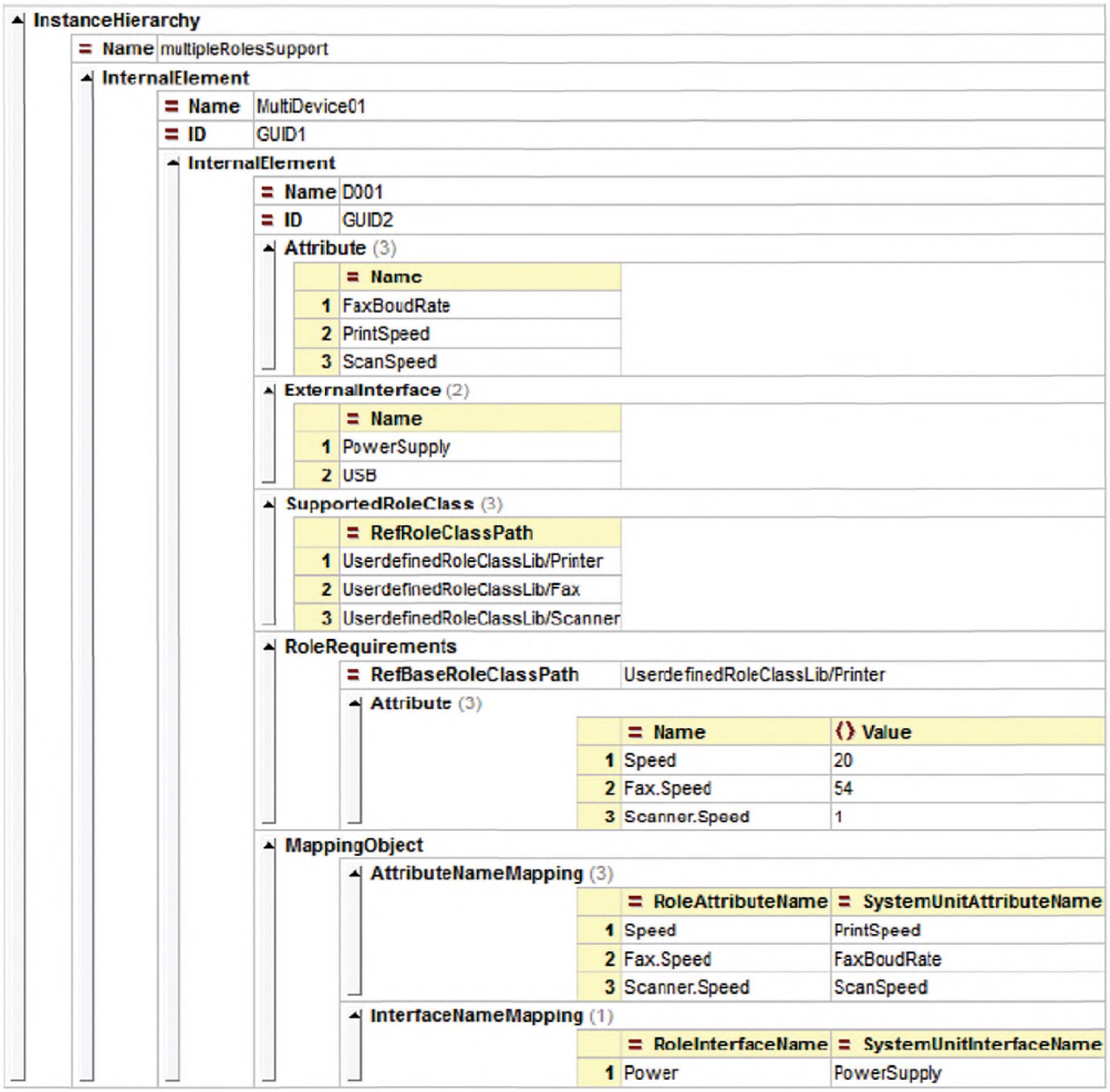
```
<InstanceHierarchy Name="Example_InstanceHierarchy">
  <InternalElement Name="Example_Plant" ID="GUID1">
    <InternalElement Name="Resources" ID="GUID2">
      <InternalElement Name="C1" ID="GUID3">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Resource"/>
      </InternalElement>
      <InternalElement Name="TT1" ID="GUID4">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Resource"/>
      </InternalElement>
      <InternalElement Name="RB1" ID="GUID5">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Resource"/>
      </InternalElement>
      <InternalElement Name="C2" ID="GUID6">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Resource"/>
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Resource"/>
    </InternalElement>
    <InternalElement Name="Processes" ID="GUID7">
      <InternalElement Name="Transport1" ID="GUID8">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Process"/>
      </InternalElement>
      <InternalElement Name="Turn1" ID="GUID9">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Process"/>
      </InternalElement>
      <InternalElement Name="Transport2" ID="GUID10">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Process"/>
      </InternalElement>
      <InternalElement Name="Assemble1" ID="GUID11">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Process"/>
      </InternalElement>
      <InternalElement Name="Transport3" ID="GUID12">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Process"/>
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Process"/>
    </InternalElement>
    <InternalElement Name="Products" ID="GUID13">
      <InternalElement Name="Car without wheels1" ID="GUID14">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Product"/>
      </InternalElement>
      <InternalElement Name="Wheels1" ID="GUID15">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Product"/>
      </InternalElement>
      <InternalElement Name="Car with wheels1" ID="GUID16">
        <ExternalInterface Name="PPR" RefBaseClassPath="AutomationMLInterfaceClassLib/.../PPRConnector"/>
        <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Product"/>
      </InternalElement>
      <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Product"/>
    </InternalElement>
    <InternalLink Name="C1_T1" RefPartnerSideA="GUID3:PPR" RefPartnerSideB="GUID8:PPR"/>
    <InternalLink Name="TT1_Tu1" RefPartnerSideA="GUID4:PPR" RefPartnerSideB="GUID9:PPR"/>
    <InternalLink Name="RB1_A1" RefPartnerSideA="GUID5:PPR" RefPartnerSideB="GUID11:PPR"/>
    <InternalLink Name="C2_T2" RefPartnerSideA="GUID6:PPR" RefPartnerSideB="GUID10:PPR"/>
    <InternalLink Name="C2_T3" RefPartnerSideA="GUID6:PPR" RefPartnerSideB="GUID12:PPR"/>
    <InternalLink Name="C1_CwW1" RefPartnerSideA="GUID3:PPR" RefPartnerSideB="GUID14:PPR"/>
    <InternalLink Name="TT1_CwW1" RefPartnerSideA="GUID4:PPR" RefPartnerSideB="GUID14:PPR"/>
    <InternalLink Name="RB1_CwW1" RefPartnerSideA="GUID5:PPR" RefPartnerSideB="GUID14:PPR"/>
    <InternalLink Name="RB1_W1" RefPartnerSideA="GUID5:PPR" RefPartnerSideB="GUID15:PPR"/>
    <InternalLink Name="RB1_CW1" RefPartnerSideA="GUID5:PPR" RefPartnerSideB="GUID16:PPR"/>
    <InternalLink Name="C2_CwW1" RefPartnerSideA="GUID6:PPR" RefPartnerSideB="GUID14:PPR"/>
    <InternalLink Name="C2_CW1" RefPartnerSideA="GUID6:PPR" RefPartnerSideB="GUID16:PPR"/>
    <RoleRequirements RefBaseRoleClassPath="AutomationMLBaseRoleClassLib/.../Cell"/>
  </InternalElement>
</InstanceHierarchy>
```

Рисунок А.36 — Представление иерархии экземпляров InstanceHierarchy рассматриваемого примера на языке XML

MappingObject в соответствии с МЭК 62424 (раздел А.2.10), который дает информацию об атрибуте роли (интерфейсе), о порядке ассоциации рассматриваемого экземпляра атрибута (интерфейса). Чтобы различать атрибуты различных ролей (имеющих одинаковые имена), имя роли включается в определение отображения (за исключением базовой роли, указанной в пути доступа «RefBaseRoleClassPath»). Рисунок А.37 содержит пример описания необходимых атрибутов и интерфейсов, а также порядка их отображения на атрибуты и интерфейсы экземпляры.

Рисунок А.38 содержит представление рассматриваемой структуры на языке AutomationML.

Рисунки А.39 и А.40 содержат соответствующие библиотеки ролевых классов AutomationML и их представление на языке XML.

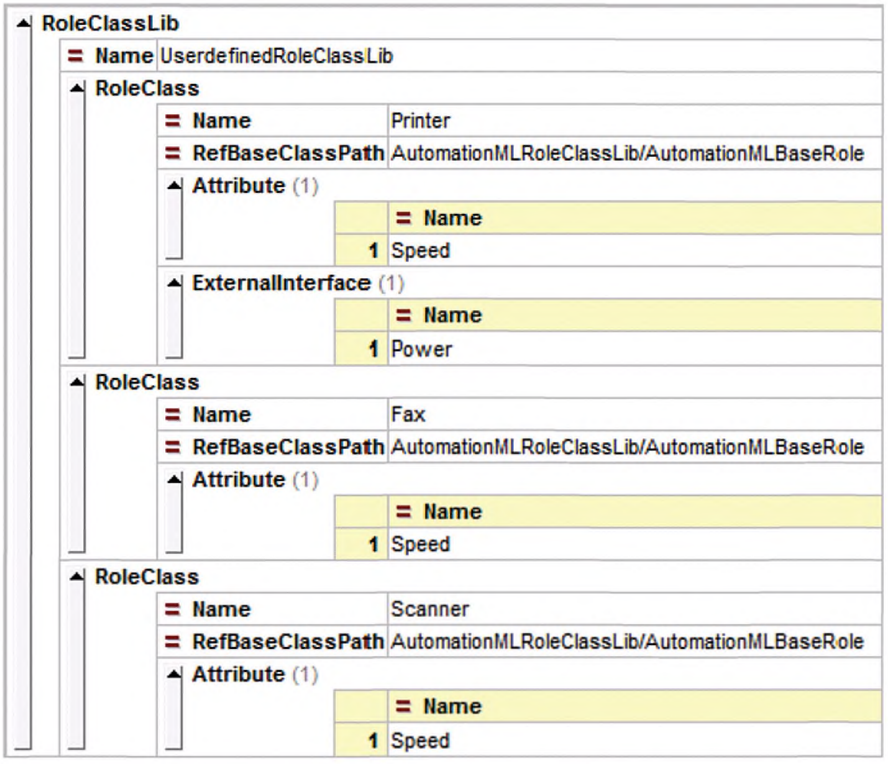


InstanceHierarchy — Иерархия экземпляров; SupportedRoleClass — Поддерживаемый ролевой класс; RefRoleClassPath — Ссылочный путь доступа к ролевому классу; UnderfinedRoleClassLib/Printer — Библиотека пользовательского ролевых классов/принтер; RoleRequirements — Требования к роли; RefBaseRoleClassPath — Ссылочный путь доступа к базовому ролевому классу; MappingObject — Объект отображения; AttributeNameMapping — Отображение имени атрибута; RoleAttributeName — Имя атрибута роли; InterfaceNameMapping — Отображение имени интерфейса; RoleInterfaceName — Имя интерфейса роли; SystemUnitInterfaceName — Имя интерфейса системной единицы; PowerSupply — Подача питания

Рисунок А.37 — Пример пользовательского экземпляра, поддерживающего несколько ролей

```
<InstanceHierarchy Name="multipleRolesSupport">
  <InternalElement Name="MultiDevice01" ID="GUID1">
    <InternalElement Name="D001" ID="GUID2">
      <Attribute Name="FaxBoudRate"/>
      <Attribute Name="PrintSpeed"/>
      <Attribute Name="ScanSpeed"/>
      <ExternalInterface Name="PowerSupply"/>
      <ExternalInterface Name="USB"/>
      <SupportedRoleClass RefRoleClassPath="UserdefinedRoleClassLib/Printer"/>
      <SupportedRoleClass RefRoleClassPath="UserdefinedRoleClassLib/Fax"/>
      <SupportedRoleClass RefRoleClassPath="UserdefinedRoleClassLib/Scanner"/>
      <RoleRequirements RefBaseRoleClassPath="UserdefinedRoleClassLib/Printer">
        <Attribute Name="Speed">
          <Value>20</Value>
        </Attribute>
        <Attribute Name="Fax.Speed">
          <Value>54</Value>
        </Attribute>
        <Attribute Name="Scanner.Speed">
          <Value>1</Value>
        </Attribute>
      </RoleRequirements>
      <MappingObject>
        <AttributeNameMapping RoleAttributeName="Speed" SystemUnitAttributeName="PrintSpeed"/>
        <AttributeNameMapping RoleAttributeName="Fax.Speed" SystemUnitAttributeName="FaxBoudRate"/>
        <AttributeNameMapping RoleAttributeName="Scanner.Speed" SystemUnitAttributeName="ScanSpeed"/>
        <InterfaceNameMapping RoleInterfaceName="Power" SystemUnitInterfaceName="PowerSupply"/>
      </MappingObject>
    </InternalElement>
  </InternalElement>
```

Рисунок А.38 — XML текст представления, поддерживающего несколько ролей на языке AutomationML



RoleClassLib — Библиотека ролевых классов; RoleClass — Ролевой класс; RefBaseClassPath — Ссылочный путь доступа к базовому классу; ExternalIInterface — Внешний интерфейс

Рисунок А.39 — Библиотека ролевых классов AutomationML, соответствующая примеру с несколькими определениями ролей

```
<RoleClassLib Name="UserdefinedRoleClassLib">
  <RoleClass Name="Printer" RefBaseClassPath="AutomationMLRoleClassLib/AutomationMLBaseRole">
    <Attribute Name="Speed"/>
    <ExternalInterface Name="Power"/>
  </RoleClass>
  <RoleClass Name="Fax" RefBaseClassPath="AutomationMLRoleClassLib/AutomationMLBaseRole">
    <Attribute Name="Speed"/>
  </RoleClass>
  <RoleClass Name="Scanner" RefBaseClassPath="AutomationMLRoleClassLib/AutomationMLBaseRole">
    <Attribute Name="Speed"/>
  </RoleClass>
</RoleClassLib>
```

Рисунок А.40 — Текст на языке XML для библиотеки ролевых классов AutomationML

Приложение В
(справочное)

Представление библиотек AutomationML на языке XML

В.1 Библиотека базового ролевых классов AutomationMLBaseRoleClassLib

```
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile xmlns:xsi="gost.e-ecolog.ru
    xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd"
    FileName="AutomationMLBaseRoleClassLib.AutomationML" SchemaVersion="2.15">
  <AdditionalInformation AutomationMLVersion="2.0" />
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>IEC SC65E WG 9</WriterName>
      <WriterID>IEC SC65E WG 9</WriterID>
      <WriterVendor>IEC</WriterVendor>
      <WriterVendorURL>www.iec.ch</WriterVendorURL>
      <WriterVersion>1.0</WriterVersion>
      <WriterRelease>1.0.0</WriterRelease>
      <LastWritingDateTime>2013-03-01</LastWritingDateTime>
      <WriterProjectTitle>Automation Markup Language Standard
        Libraries</WriterProjectTitle>
      <WriterProjectID>Automation Markup Language Standard
        Libraries</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <ExternalReference Path="../InterfaceClass
    Libraries/AutomationMLInterfaceClassLib.AutomationML"
    Alias="AutomationMLInterfaceClassLib" />
  <RoleClassLib Name="AutomationMLBaseRoleClassLib">
    <Description>Automation Markup Language base role class
      library</Description>
    <Version>2.2.0</Version>
    <RoleClass Name="AutomationMLBaseRole">
      <RoleClass Name="Group" RefBaseClassPath="AutomationMLBaseRole">
        <Attribute Name="AssociatedFacet" AttributeDataType="xs:string" />
      </RoleClass>
      <RoleClass Name="Facet" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Port" RefBaseClassPath="AutomationMLBaseRole">
        <Attribute Name="Direction" AttributeDataType="xs:string" />
        <Attribute Name="Cardinality">
          <Attribute Name="MinOccur" AttributeDataType="xs:unsignedInt" />
          <Attribute Name="MaxOccur" AttributeDataType="xs:unsignedInt" />
        </Attribute>
        <Attribute Name="Category" AttributeDataType="xs:string" />
        <ExternalInterface Name="ConnectionPoint" ID="9942bd9c-c19d-44e4-a197-
          11b9edf264e7"
          RefBaseClassPath="AutomationMLInterfaceClassLib@AutomationMLInterfaceC
          lassLib/AutomationMLBaseInterface/PortConnector" />
      </RoleClass>
      <RoleClass Name="Resource" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Product" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Process" RefBaseClassPath="AutomationMLBaseRole" />
      <RoleClass Name="Structure" RefBaseClassPath="AutomationMLBaseRole">
        <RoleClass Name="ProductStructure" RefBaseClassPath="Structure" />
        <RoleClass Name="ProcessStructure" RefBaseClassPath="Structure" />
        <RoleClass Name="ResourceStructure" RefBaseClassPath="Structure" />
      </RoleClass>
      <RoleClass Name="PropertySet" RefBaseClassPath="AutomationMLBaseRole" />
    </RoleClass>
  </RoleClassLib>
</CAEXFile>
```

B.2 Библиотека классов интерфейса AutomationMLInterfaceClassLib

```
<?xml version="1.0" encoding="utf-8"?>
<CAEXFile xmlns:xsi="gost.e-ecolog.ru
xsi:noNamespaceSchemaLocation="CAEX_ClassModel_V2.15.xsd"
FileName="AutomationMLInterfaceClassLib.AutomationML" SchemaVersion="2.15">
  <AdditionalInformation AutomationMLVersion="2.0" />
  <AdditionalInformation>
    <WriterHeader>
      <WriterName>IEC SC65E WG 9</WriterName>
      <WriterID>IEC SC65E WG 9</WriterID>
      <WriterVendor>IEC</WriterVendor>
      <WriterVendorURL>www.iec.ch</WriterVendorURL>
      <WriterVersion>1.0</WriterVersion>
      <WriterRelease>1.0.0</WriterRelease>
      <LastWritingDateTime>2013-03-01</LastWritingDateTime>
      <WriterProjectTitle>Automation Markup Language Standard
        Libraries</WriterProjectTitle>
      <WriterProjectID>Automation Markup Language Standard
        Libraries</WriterProjectID>
    </WriterHeader>
  </AdditionalInformation>
  <InterfaceClassLib Name="AutomationMLInterfaceClassLib">
    <Description>Standard Automation Markup Language Interface Class
      Library</Description>
    <Version>2.2.0</Version>
    <InterfaceClass Name="AutomationMLBaseInterface">
      <InterfaceClass Name="Order" RefBaseClassPath="AutomationMLBaseInterface">
        <Attribute Name="Direction" AttributeDataType="xs:string" />
      </InterfaceClass>
      <InterfaceClass Name="PortConnector"
        RefBaseClassPath="AutomationMLBaseInterface" />
      <InterfaceClass Name="InterlockingConnector"
        RefBaseClassPath="AutomationMLBaseInterface" />
      <InterfaceClass Name="PPRConnector"
        RefBaseClassPath="AutomationMLBaseInterface" />
      <InterfaceClass Name="ExternalDataConnector"
        RefBaseClassPath="AutomationMLBaseInterface">
        <Attribute Name="refURI" AttributeDataType="xs:anyURI" />
        <InterfaceClass Name="COLLADAInterface"
        RefBaseClassPath="ExternalDataConnector" />
        <InterfaceClass Name="PLCopenXMLInterface"
        RefBaseClassPath="ExternalDataConnector" />
      </InterfaceClass>
      <InterfaceClass Name="Communication"
        RefBaseClassPath="AutomationMLBaseInterface">
        <InterfaceClass Name="SignalInterface" RefBaseClassPath="Communication"
            />
        </InterfaceClass>
    </InterfaceClass>
  </InterfaceClassLib>
</CAEXFile>
```

Приложение ДА
(справочное)

Сведения о соответствии ссылочных международных стандартов национальным стандартам

Таблица ДА.1

Обозначение ссылочного международного стандарта	Степень соответствия	Обозначение и наименование соответствующего национального стандарта
IEC 62424	—	*
IEC 62714-1	—	*
IEC 62714-2	—	*
ISO/IEC 9834-8	IDT	ГОСТ Р ИСО/МЭК 9834-8:2011 «Информационная технология. Взаимосвязь открытых систем. Процедуры работы уполномоченных по регистрации ВОС. Часть 8. Создание, регистрация универсально уникальных идентификаторов (УУИд) и их использование в качестве компонентов идентификатора объекта АСН.1»
ISO/PAS 17506	—	*

* Соответствующий национальный стандарт отсутствует.

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

- IDT — идентичный стандарт.

Библиография

1 IEC 60027 (all parts) Letter symbols to be used in electrical technology
2 Обозначения буквенные, применяемые в электротехнике (все части IEC 60027)
3 IEC 62264-1 Enterprise-control system integration — Part 1: Models and terminology (Интеграция систем управления предприятием. Часть 1. Модели и терминология)
4 IEC 62714-2 Engineering data exchange format for use in industrial automation systems engineering — Automation Markup Language — Part 2: Role class libraries (Формат обмена техническими данными, используемый при проектировании промышленных автоматизированных систем. Язык разметки автоматизации. Часть 2. Библиотеки ролевых классов)
5 ISO 80000-1 Quantities and units — Part 1: General (Величины и единицы. Часть 1. Общие положения)

УДК 658.52.011.56:006.354 ОКС 25.040.40, 35.060, 35.240.50

Ключевые слова: системы промышленной автоматизации, интеграция, формат обмена инженерными данными, стандартизованный формат обмена данными AutomationML

Редактор *А.Е. Петросян*
Технический редактор *В.Ю. Фотиева*
Корректор *С.В. Смирнова*
Компьютерная верстка *А.А. Ворониной*

Сдано в набор 12.12.2016. Подписано в печать 30.01.2017. Формат 60×84 1/8. Гарнитура Ариал.
Усл. печ. л. 8,37. Уч.-изд. л. 7,55. Тираж 28 экз. Зак. 246.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

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

gost.e-ecolog.ru