ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ГОСТ Р 71808— 2024**

# Цифровая промышленность

# УНИФИЦИРОВАННАЯ АРХИТЕКТУРА OPC

## Часть 3

## Модель адресного пространства

**(IEC 62541-3:2020, NEQ)**

**Издание официальное**

**Москва**
**Российский институт стандартизации**
**2025**

gost.e-ecolog.ru

## Предисловие

1 РАЗРАБОТАН Ассоциацией «Цифровые инновации в машиностроении» и Федеральным государственным бюджетным учреждением «Российский институт стандартизации» (ФГБУ «Институт стандартизации»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 306 «Измерения, управление и автоматизация в промышленных процессах»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 29 ноября 2024 г. № 1804-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений международного стандарта МЭК 62541-3—2020 «Унифицированная архитектура OPC. Часть 3. Модель адресного пространства» (IEC 62541-3:2020 «OPC unified architecture — Part 3: Address space model», NEQ)

5 ВВЕДЕН ВПЕРВЫЕ

*Правила применения настоящего стандарта установлены в статье 26 Федерального закона от 29 июня 2015 г. № 162-ФЗ «О стандартизации в Российской Федерации». Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок — в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Оформление. ФГБУ «Институт стандартизации», 2025

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, сокращения и обозначения . . . . . 2
3.1 Термины и определения . . . . . 2
3.2 Сокращения . . . . . 2
3.3 Обозначения . . . . . 2
4 Концепции адресного пространства . . . . . 4
4.1 Модель объекта . . . . . 4
4.2 Модель узла . . . . . 4
4.3 Переменные . . . . . 5
4.4 Узлы определения типов . . . . . 6
4.5 Модель события . . . . . 7
4.6 Методы . . . . . 8
4.7 Роли . . . . . 8
5 Стандартные классы узла . . . . . 10
5.1 Обзор . . . . . 10
5.2 Базовый класс узла . . . . . 10
5.3 Класс узла типа ссылки . . . . . 13
5.4 Представление класса узла . . . . . 16
5.5 Объекты . . . . . 18
5.6 Переменные . . . . . 20
5.7 Метод класса узла . . . . . 21
5.8 Типы данных . . . . . 21
6 Модель типа для типов объекта и типов переменных . . . . . 23
6.1 Определения . . . . . 23
6.2 Создание подтипов типов объектов и типов переменной . . . . . 24
7 Стандартные типы ссылок . . . . . 24
7.1 Общие положения . . . . . 24
7.2 Тип ссылки ссылок . . . . . 24
7.3 Тип ссылки иерархических ссылок . . . . . 24
7.4 Тип неиерархической ссылки . . . . . 24
7.5 Тип ссылки агрегации . . . . . 24
7.6 Тип ссылки наличия компонента . . . . . 24
7.7 Тип ссылки наличия свойства . . . . . 24
7.8 Тип ссылки наличия подтипа . . . . . 25
7.9 Тип ссылки организации . . . . . 25
7.10 Тип ссылки наличия правила моделирования . . . . . 25
7.11 Тип ссылки наличия типа определения . . . . . 25
7.12 Тип ссылки наличия кодирования . . . . . 25
7.13 Генерация события . . . . . 25
7.14 Тип ссылки «Всегда генерирует событие» . . . . . 25
7.15 Наличие источника события . . . . . 26
7.16 Наличие уведомителя . . . . . 26
8 Стандартные типы данных . . . . . 26
8.1 Общие положения . . . . . 26
8.2 Тип идентификатора . . . . . 26
8.3 Объединение . . . . . 26
8.4 Тип ограничения доступа . . . . . 26
8.5 Тип уровня доступа . . . . . 27
8.6 Рамка записи атрибута . . . . . 27
Библиография . . . . . 28

## Введение

Настоящий стандарт является третьей частью серии стандартов «Цифровая промышленность. Унифицированная архитектура OPC». Он является нормативным представлением одного из инструментов по управлению процессами цифровой трансформации производств, основанных на интеграции и интероперабельности автоматизированных систем, технологического оборудования, промышленных роботов, средств оснащения, контроллеров и датчиков. Являясь мета-моделью унифицированной архитектуры открытой распределенной системы, на которой основаны информационные модели, настоящий стандарт определяет адресное пространство унифицированной архитектуры OPC (OPC UA) и его объекты.

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Цифровая промышленность**

# УНИФИЦИРОВАННАЯ АРХИТЕКТУРА OPC

**Часть 3**

**Модель адресного пространства**

Digital industry.
OPC unified architecture.
Part 3. Address space model

**Дата введения — 2025—02—01**

## 1 Область применения

Настоящий стандарт определяет адресное пространство унифицированной архитектуры OPC (OPC UA) и его объекты. Данный стандарт является мета-моделью OPC UA, на которой основаны информационные модели OPC UA.

## 2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 59799 Умное производство. Модель эталонной архитектуры индустрии 4.0 (RAMI 4.0)

ГОСТ Р 70988 Система стандартов в цифровой промышленности. Основные положения. Общие требования к системе

ГОСТ Р 71806 Цифровая промышленность. Унифицированная архитектура OPC. Часть 1. Обзор и концепции

ГОСТ Р 71809 Цифровая промышленность. Унифицированная архитектура OPC. Часть 4. Сервисы

ГОСТ Р 71810 Цифровая промышленность. Унифицированная архитектура OPC. Часть 5. Информационная модель

ГОСТ Р 71811 Цифровая промышленность. Унифицированная архитектура OPC. Часть 6. Сопоставления

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

**Издание официальное**

## 3 Термины, определения, сокращения и обозначения

### 3.1 Термины и определения

В данном стандарте применены следующие термины с соответствующими определениями:

3.1.1 **декларация экземпляра** (instance declaration): Узел, использующийся в сложных узлах определения типов с целью раскрытия их сложной структуры.

Примечание — Данный пример используется в определении типа.

3.1.2 **идентификатор типа данных** (data type Id)**:** Идентификатор узла в узле типа данных.

Примечание — См. [1].

3.1.3 **иерархическая ссылка** (hierarchical reference): Ссылка, которая используется для построения иерархий в адресном пространстве.

Примечание — Все иерархические типы ссылок выводятся из иерархических ссылок.

3.1.4 **переменная данных** (data variable): Переменная, прямо или косвенно представляющая значение объекта для сложных переменных, где переменные всегда являются целевым узлом ссылки на наличие компонента.

3.1.5 **правило моделирования** (modelling rule): Метаданные декларации экземпляра, определяющие, как декларация экземпляра будет использована для конкретизации, а также определяющие правила выделения подтипа для декларации экземпляра.

Примечание — См. ГОСТ Р 71809.

3.1.6 **свойство** (property): Переменная, являющаяся целевым узлом для ссылки на наличие компонента.

Примечание — Свойства описывают характеристики узла.

3.1.7 **тип данных** (data type): Пример узла типа данных, используемого вместе с атрибутом рейтинга значения для определения типа данных переменной.

3.1.8 **тип переменной** (variable type): Узел, представляющий тип описания для переменной.

3.1.9 **тип события** (event type): Узел типа события, представляющий определение типа события.

3.1.10 **узел источника** (source node): Узел, у которого есть ссылка на другой узел.

3.1.11 **узел определения типов** (type definition node): Узел, использующийся для определения типа другого узла.

Примечание — Тип объекта и узлы типов переменной являются узлами определения типа.

3.1.12 **целевой узел** (target node): Узел, на который ссылается другой узел.

### 3.2 Сокращения

В настоящем стандарте приведены следующие сокращения:

UA — унифицированная архитектура (Unified Architecture);

UML — открытый протокол для передачи сообщений между компонентами системы (Advanced Message Queuing Protocol) (Unified Modeling Language);

URI — программный интерфейс приложения (Application Programming Interface) (Uniform Resource Identifier);

W3C — модель компонентного объекта (Component Object Model) (World Wide Web Consortium);

XML — доступ к данным (Data Access) (Xetensible Markup Language).

### 3.3 Обозначения

#### 3.3.1 Условные обозначения для рисунков адресного пространства

На рисунке 1 показан бланк диаграмм узла адресного пространства с используемыми условными обозначениями.

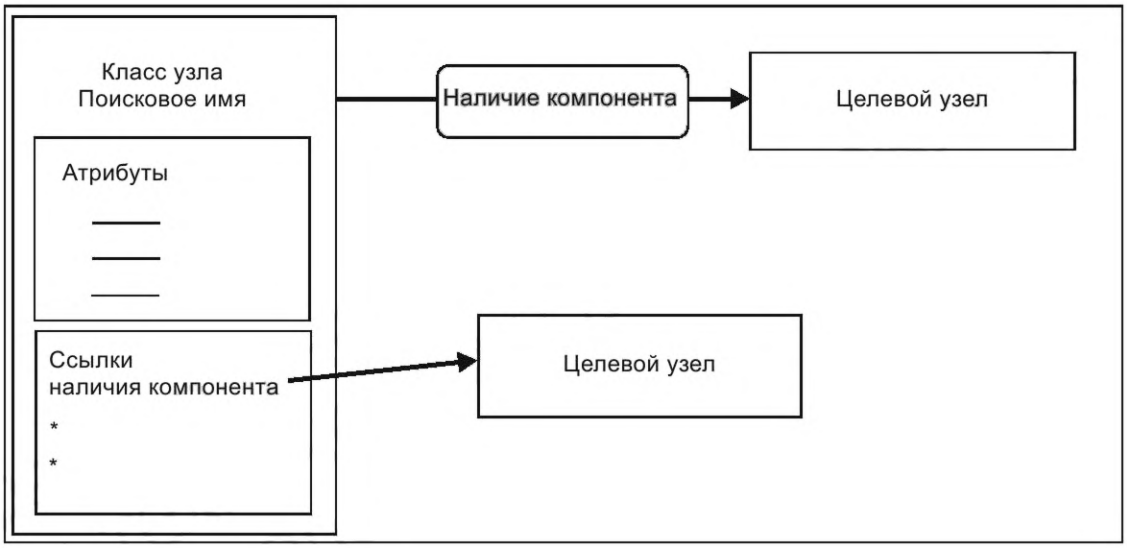


Рисунок 1 — Диаграммы узла адресного пространства

На рисунке 1 прямоугольники представляют узлы. Прямоугольники — узлы могут иметь заголовок, состоящий из одной или двух строк текста. При использовании двух строк первая текстовая строка прямоугольника идентифицирует класс узла, а вторая строка содержит поисковое имя. Когда используется одна строка, она содержит поисковое имя.

Прямоугольники — узлы могут содержать блоки, используемые для определения их атрибутов и ссылок. Конкретные названия в этих блоках определяют конкретные атрибуты и ссылки.

Заштрихованный прямоугольник с закругленными углами и стрелками, проходящими через него, представляет собой ссылку. Стрелка, которая проходит через них, начинается в узле источника и указывает на целевой узел.

**3.3.2 Условные обозначения для определения классов узла**

В разделе 4 определены классы узлов адресного пространства. В таблице 1 описан формат таблиц, используемых для определения классов узлов.

Таблица 1 — Таблица условных обозначений классов узлов

| Имя | Использование | Тип данных | Описание |
|---|---|---|---|
| Атрибуты | | | |
| «Имя атрибута» | «О» или «Н» | Тип данных атрибута | Определяет атрибут |
| Ссылки | | | |
| «Имя ссылки» | «1», «0...1» или «0…» | Не используется | Описывает использование ссылки с помощью класса узла |
| Стандартные свойства | | | |
| «Имя свойства» | «О» или «Н» | Тип данных свойства | Определяет свойство |

Графа «Имя» содержит имя атрибута, имя типа ссылки, используемые для создания ссылки, или имя свойства, на которое ссылаются с помощью ссылки наличия свойства.

Графа «Использование» определяет, является ли атрибут или свойство обязательным (О) или необязательным (Н). В случае, если это обязательно, атрибут или свойство должны существовать для каждого узла класса узла. Для ссылок он указывает количество элементов. Могут применяться следующие значения:

- «0.*» указывает, что ограничений нет, то есть ссылка не обязательно должна быть предоставлена;
- «0.1» указывает, что ссылка предоставляется не более одного раза;
- «1» указывает, что ссылка должна быть предоставлена ровно один раз.

Графа «Тип данных» содержит название типа данных атрибута или свойства. Он не используется для ссылок.

Графа «Описание» содержит описание атрибута, ссылки или свойства.

Только данный стандарт может определять атрибуты. Все атрибуты класса узла могут быть расширены другими частями серии стандартов OPC UA.

Данный стандарт определяет типы ссылок. Допускается, что типы ссылок могут быть установлены сервером или клиентом с использованием службы управления узлом, указанных в ГОСТ Р 71809. Данный стандарт определяет свойства, представляемые своим именем, которое сопоставляется с поисковым именем, имеющим индекс пространства имени <0>, который представляет пространство имени для OPC UA.

Графа «Использование» (необязательный или обязательный) не подразумевает конкретного правила моделирования для свойств.

# 4 Концепции адресного пространства

## 4.1 Модель объекта

Основная цель адресного пространства OPC UA — дать серверу стандартный способ представления объектов клиентам. Модель объекта OPC UA определяет объекты в терминах переменных и методов. Она позволяет выражать отношения к другим объектам.

Элементы данной модели представлены в адресном пространстве в виде узлов. Каждый узел назначен классом узла, и каждый класс узла представляет собой отдельный элемент модели объекта. Раздел 4 определяет классы узла, используемые для представления данной модели.

## 4.2 Модель узла

### 4.2.1 Общие положения

Набор объектов и связанная с ними информация, доступ к которой сервер OPC UA дает клиентам, называется его адресным пространством. Модель для объектов определяется моделью объекта OPC UA (см. 4.1).

Объекты и их компоненты представлены в адресном пространстве в виде набора узлов, описываемых атрибутами и взаимосвязанных ссылками. На рисунке 2 показан бланк представления атрибутов модели узла и ссылок на базовые значения.

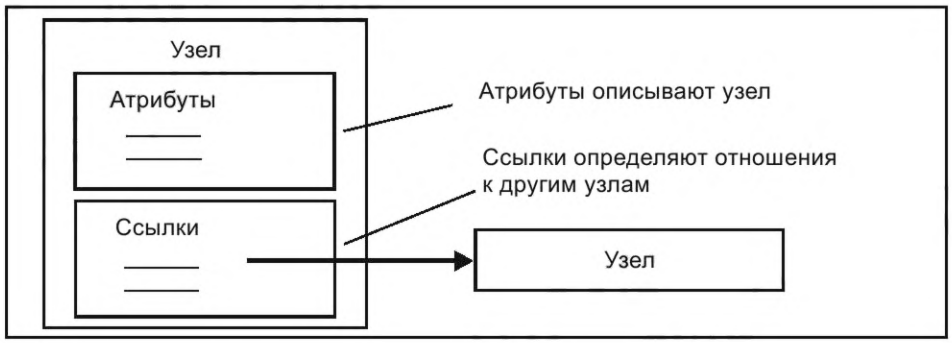


Рисунок 2 — Модель узла адресного пространства

### 4.2.2 Классы узла

Когда узел определен в адресном пространстве, его класс определен заданными атрибутами и их базовыми значениями. Представление атрибутов дано в 4.2.3, а базовые значения в 4.2.4.

Раздел 4 определяет классы узлов для адресного пространства OPC UA. Данная совокупность классов узлов является метаданными адресного пространства. Каждый узел в адресном пространстве представляет собой экземпляр одного из классов данного узла. Клиентам и серверам не разрешается вводить определения классов узлов или расширять имеющиеся определения данных классов узлов.

**4.2.3 Атрибуты**

Атрибуты — элементы данных, описывающие узлы. Клиенты могут получить доступ к значениям атрибутов с помощью служб «Читать», «Записывать», «Запрос» и «Подписка/Элемент» мониторинга. Эти службы определены в ГОСТ Р 71809.

Атрибуты являются элементарными компонентами классов узла. Каждое определение атрибута состоит из идентификатора атрибута (идентификаторы атрибутов можно найти в ГОСТ Р 71811), имени, описания, типа данных и обязательного/необязательного индикатора. Набор атрибутов, определенных для каждого класса узла, не должен расширяться клиентами или серверами.

Когда узел конкретизируется в адресном пространстве, предоставляются значения атрибутов класса узла.

**4.2.4 Ссылки**

Ссылки на базовые значения атрибутов используются для связи узлов друг с другом. Доступ к ним определен службами просмотра и запроса, представленными в ГОСТ Р 71809.

Ссылки являются фундаментальными компонентами узлов. В отличие от атрибутов ссылки определяются как узлы типов ссылки. Узлы типов ссылки представлены в адресном пространстве с помощью класса узла типов ссылки с помощью модели (см. 4.2).

Узел, содержащий ссылку, называется узлом источника, а узел, на который ссылаются, называется целевым узлом. Комбинация из узла источника, типа ссылки и целевого узла применяется в службах OPC UA для уникальной идентификации ссылок. Каждый узел может ссылаться на другой узел с тем же самым типом ссылки лишь один раз. Любые подтипы типов ссылок при идентификации ссылок считаются равными конкретным типам ссылок. На рисунке 3 представлен бланк модели ссылки.

Допускается, что целевой узел ссылки может находиться в том же самом адресном пространстве или в адресном пространстве другого сервера OPC UA. Целевые узлы, расположенные на других серверах, идентифицируются в службах OPC UA, используя комбинацию имени удаленного сервера и идентификатора, присвоенного удаленным сервером узлу.

OPC UA не требует существования целевого узла, поэтому ссылки могут указывать на несуществующий узел.

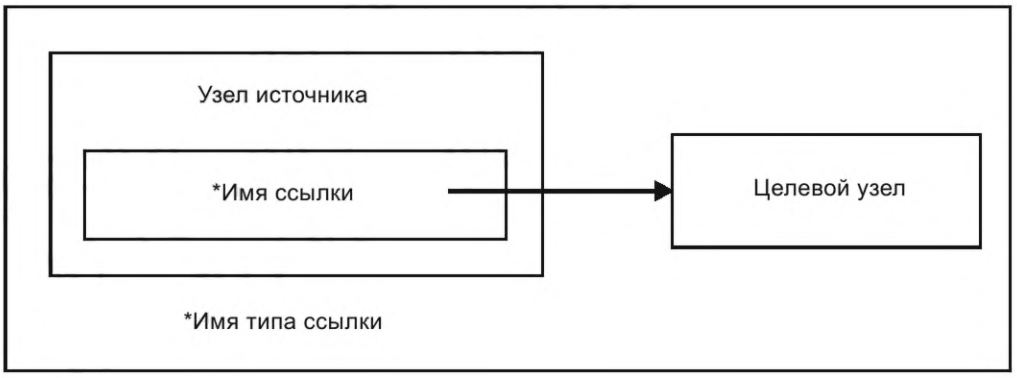


Рисунок 3 — Модель ссылки

**4.3 Переменные**

**4.3.1 Общие положения**

Переменные используются для представления значений. Определены два типа переменных: свойства и переменные данных. Они различаются типом представляемых данных и тем, могут ли они содержать другие переменные.

**4.3.2 Свойства**

Свойства — определяемые сервером характеристики объектов, переменных данных и других узлов. Свойства отличаются от атрибутов тем, что характеризуют то, что представляет собой узел, например устройство или заказ дополнительного объекта. Свойства определяются сервером, а атрибуты

определяют дополнительные метаданные, которые конкретизируются для всех узлов из класса узлов. Атрибуты являются общими для всех узлов в классе узла.

Например, атрибут определяет тип данных переменных, а свойство может быть использовано для указания инженерной единицы некоторых переменных. Для предотвращения рекурсии самим свойствам не разрешается определять свойства. Для строгости идентификации свойства свойство поискового имени должно быть уникальным в контексте узла, содержащего свойства (подробнее см. в 5.6.2).

Узел и его свойства всегда должны находиться на одном сервере.

#### 4.3.3 Переменные данных

Переменные данных представляют собой содержание объекта. Например, можно определить файловый объект, содержащий поток байтов. Поток байтов может быть определен как переменная данных, представляющая собой массив байтов. Свойства могут быть использованы для отображения времени создания и владельца файла объект.

Если переменные данных являются сложными, они могут обладать дополнительными переменными данных. В таком случае их переменные данных должны быть элементами их сложных определений. В таком случае эта сложная переменная может определять внутри себя ссылку на наличие компонента к индивидуальным значениям данных, из которых она состоит.

### 4.4 Узлы определения типов

#### 4.4.1 Общие положения

Серверам OPC UA следует предоставлять определения типов для объектов и переменных. Ссылка на наличие определения типа используется для связи экземпляра с его определением типа, представленным узлом определения типа. ГОСТ Р 71810 определяет тип базового объекта. Если специализированная информация о типах недоступна, то у сервера есть возможность использовать базовый тип свойства и тип базовой переменной данных. Объекты и переменные наследуют атрибуты, заданные их узлом определения типа (подробнее см. в 6.2.2).

Для общих клиентов должны быть обеспечены узлы определения типа. Данные узлы определения типа могут существовать на другом сервере.

Пример, проиллюстрированный на рисунке 4, представляет бланк использования ссылки на наличие определения типа. В данном примере в адресном пространстве в качестве переменной данных представлен параметр уставки «SP». Эта переменная данных является частью объекта, не показанного на рисунке 4.

Для обеспечения общего определения заданного значения, которое может быть использовано другими объектами, применяется специализированный тип переменной.

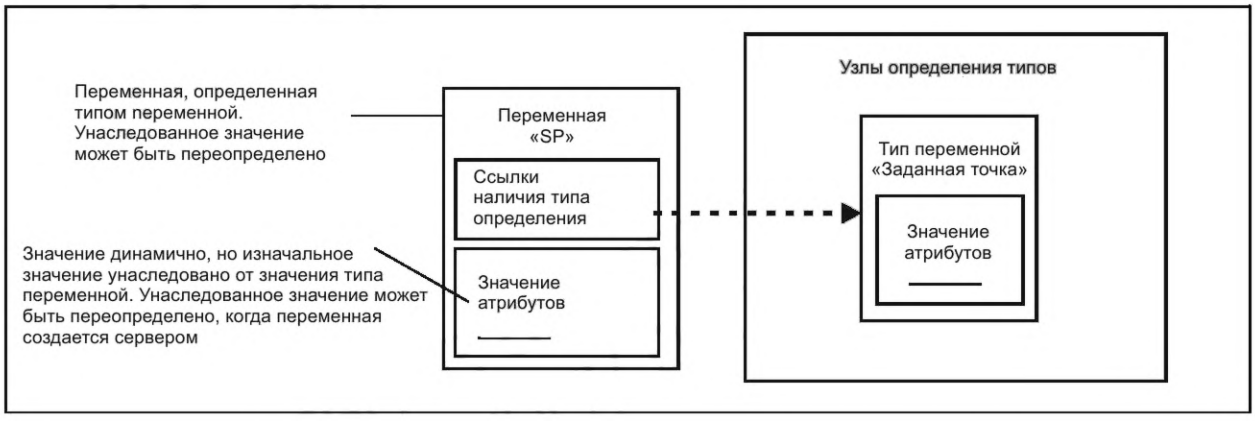


Рисунок 4 — Пример переменной, определенной типом переменной

#### 4.4.2 Комплекс узлов определения типов и их деклараций экземпляра

Узлы определения типов могут быть сложными. Сложные узлы определения типов также определяют ссылки на другие узлы как часть определения типа. Правила моделирования, определенные

в 3.1.5, устанавливают, как эти узлы обрабатываются при создании экземпляра типа определения (см. рисунок 5).

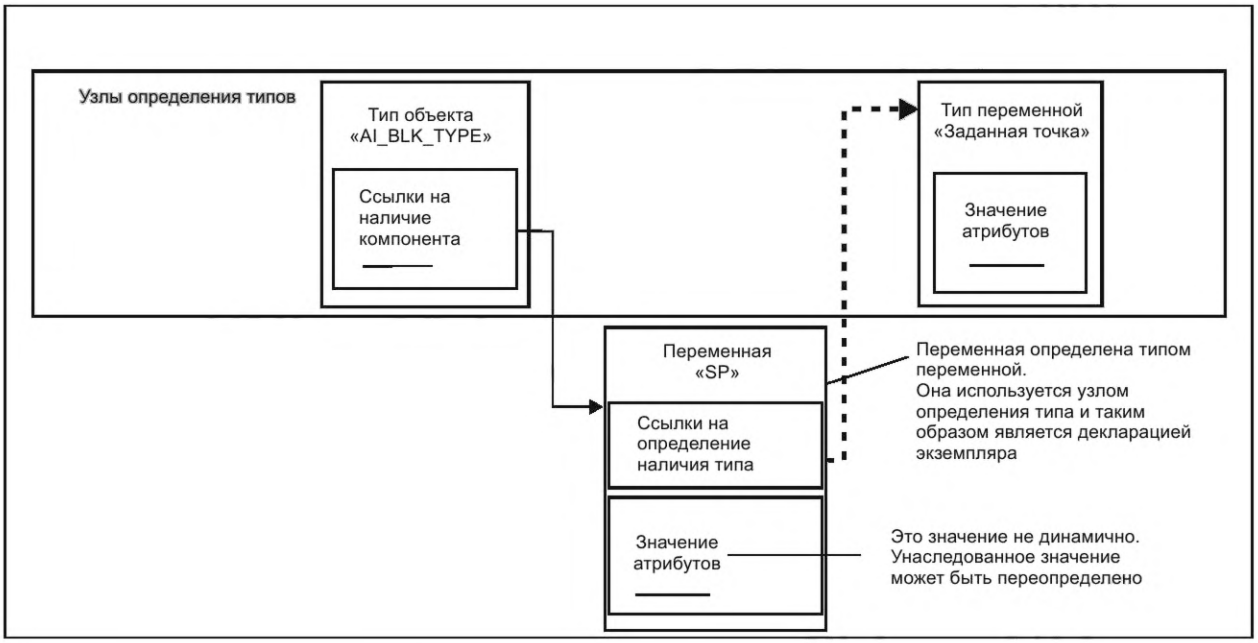


Рисунок 5 — Пример комплексного определения типа

Такой подход широко используется в объектно-ориентированных языках программирования, в которых переменные класса определяются как экземпляры других классов. Когда класс конкретизируется, каждая переменная также конкретизируется, но для содержащего класса конструктор может переопределять значения компонентов, установленные классом компонента.

**4.4.3 Создание подтипов**

Данный стандарт позволяет определения типов разделять на подтипы. Правила создания подтипов определены в разделе 5. Подтипы отражают структуру, определяемую их супертипом, но могут добавлять дополнительные характеристики. Производителю допускается создавать новый тип переменной, который является целевым узлом для ссылки на наличие подтипа из оригинального типа переменной, добавив к нему новое свойство.

Определение ограничений доступа узла представлено в 7.4, а типы уровня доступа в 7.5

### 4.5 Модель события

#### 4.5.1 Общие положения

Модель событий определяет систему событий общего назначения, которую можно использовать на самых разных вертикальных рынках.

События представляют собой специфические переходные явления. Изменения конфигурации системы и системные ошибки являются примерами событий. Атрибут уведомителя событий этих узлов определяет, разрешает ли узел подписку на события. Подписки на событие используют службы подписки/элементов контроля, определенные в ГОСТ Р 71809, для подписки на использование уведомителей событий узла.

Любой сервер OPC UA, поддерживающий генерацию событий, должен показывать, как минимум, один узел в качестве уведомителя события. Для этой цели используется объект сервера, определенный в ГОСТ Р 71810.

**4.5.2 Типы событий**

Каждое событие имеет свой собственный тип событий. Сервер может поддерживать многие типы. Данный стандарт определяет тип базового события, из которого извлекаются все остальные типы событий. Ожидается, что другие сопутствующие спецификации будут определять дополнительные типы событий, производные от базовых типов, определенных в данном стандарте.

Типы событий, поддерживаемые сервером, отображаются в адресном пространстве сервера. Типы событий представляются как типы объекта в адресном пространстве и не имеют связанного с ними особого класса узла. ГОСТ Р 71810 определяет, как сервер раскрывает типы событий.

Типы событий, определенные в данном стандарте, задаются в абстрактном виде. События этих типов событий отображаются только через подписку. Типы событий существуют в адресном пространстве, чтобы позволить клиентам обнаружить тип события. Данная информация используется клиентом при создании подписок на события и работе с ними. Типы событий, определенные другими частями серии стандартов OPC UA или дополнительными спецификациями, а также особые для сервера типов событий могут быть определены как не абстрактные; поэтому экземпляры этих типов событий могут быть видны в адресном пространстве, хотя события этих типов событий также доступны через механизмы уведомлений о событиях.

#### 4.5.3 Категоризация событий

События допустимо делить на категории, создав новые типы событий, являющиеся подтипами существующих типов событий, но не расширяющие существующий тип. Они используются только для идентификации события в качестве нового типа события. Допускается группировка источников событий с использованием типов ссылки событий. Например, сервер может определять объекты в адресном пространстве, которые представляют события, относящиеся к физическим устройствам, или области событий оборудования либо функций, содержащихся на сервере. Ссылки на события используются для выделения источников событий, определяющих физические устройства, от источников событий, определяющих функциональные возможности сервера. Ссылки могут использоваться для группировки физических устройств или функций сервера в иерархические области событий. В некоторых случаях источник события может быть классифицирован как устройство и функция сервера. В этом случае будут установлены два вида связи.

Клиентам допускается выбирать категорию или категории событий, определив контент-фильтры, которые включают в себя термины, определяющие тип какого-либо события или группирование источников события. Эти два механизма позволяют разделять одно событие на категории несколькими способами. Клиент может получить связанные с физическим устройством данные о всех событиях или о всех отказах конкретного устройства.

### 4.6 Методы

Методы — «легкие» функции, область действия которых ограничена владением объектом (см. примечание), похожим на методы класса в объектно-ориентированном программировании или владением типом объекта, похожим на статические методы класса. Методы вызываются клиентом, переходят к завершению на сервере и возвращают результат клиенту.

Примечание — Владение объекта или типа объекта указано в вызове службы при вызове метода [2].

Методы могут влиять на состояние владения объектом, но у них нет своего собственного явного состояния. Они считаются не обладающими никаким состоянием. Методы могут иметь различное количество входных и возвращаемых аргументов. Каждый метод описывается узлом метода класса узла. Этот узел содержит метаданные, которые идентифицируют аргументы метода и описывают его использование.

Методы вызываются с помощью службы вызовов, определенной в ГОСТ Р 71809.

### 4.7 Роли

#### 4.7.1 Обзор

Роль — функция, которую берет на себя клиент, получая доступ к серверу. Роли используются для разделения аутентификации (определения клиента) и авторизации (определения того, что клиенту разрешено делать). Разделяя эти задачи, серверы дают управление службам идентификационными данными пользователей и учетными записями, в то время как сервер управляет только разрешениями на своих узлах, назначенных на какие-либо роли.

Набор поддерживаемых сервером ролей публикуется в виде компонентов объекта ролей, определенного в ГОСТ Р 71810. Серверы должны определять базовый набор ролей и позволять клиентам конфигурации добавлять системные роли.

Когда создается сеанс, сервер должен определить, какие роли предоставляются этому сеансу. Настоящий стандарт определяет правила отображения, которые могут поддерживаться серверами. В

дополнение к стандартным правилам или вместо них на серверах могут также использоваться правила отображения, зависящие от поставщиков.

Идентификационные сопоставления приложений основаны на URI Приложении, указанном в сертификате клиента. Идентификация приложения может быть применена только в том случае, если клиент докажет наличие доверенного сертификата, используя его для создания защищенного канала или предоставив подпись в активацию сеанса (см. ГОСТ Р 71810).

Идентификационные сопоставления конечных точек основаны на URL, используемом для подключения к серверу. ГОСТ Р 71810 определяет объекты, методы и типы данных, используемые для представления и управления этими правилами отображения в адресном пространстве.

**4.7.2 Оценка разрешений с ролями**

Когда клиент пытается получить доступ к узлу, сервер проходит через список ролей, предоставленных для сеанса, и логически распределяет разрешения роли на узле. В объекте ролей адресного пространства сервера появляются роли.

Каждая роль имеет определенные правила отображения, которые проявляются как свойства объекта ролей (см. ГОСТ Р 71810). В таблице 2 даны примеры применения стандартных правил отображения для определения ролей, к которым сеанс имеет доступ.

Таблица 2 — Пример ролей

| Роль | Правила отображения | Описание |
|---|---|---|
| Аноним | Идентификаторы = Аноним<br>приложения =<br>конечные точки = | Определяющее роль правило отображения учетных данных применяется к анонимным пользователям |
| Подтвержденный пользователь | Идентификаторы = Подтвержденный пользователь<br>Приложения =<br>конечные точки = | Определяющее роль правило отображения учетных данных применяется к подтвержденным пользователям |
| Оператор 1 | Идентификаторы = пользователь с именем «Джо»<br>приложения = urn:<br>станция оператора 1<br>конечные точки = | Правило отображения учетных данных, определяющее имеющих доступ к роли конкретных пользователей с правилом приложения, которое ограничивает доступ к единственному клиентскому приложению |
| Оператор 2 | Идентификаторы = пользователи с именами «Джо» или «Энн»<br>приложения = urn:<br>станция оператора 2<br>конечные точки = | Правило отображения учетных данных, определяющее имеющих доступ к роли конкретных пользователей с правилом приложения, которое ограничивает доступ к единственному клиентскому приложению |
| Администратор | Идентификаторы = пользователь с именем «корень»<br>приложения =<br>конечные точки = | Правило отображения учетных данных, которое определяет конкретных пользователей, имеющих доступ к роли |
| Администратор | Идентификаторы = пользователь с именем «корень»<br>приложения =<br>конечные точки = opc.tcp://127.0.0.1:48000 | Правило отображения учетных данных, определяющее имеющих доступ к роли конкретных пользователей, когда они соединяются через определенную конечную точку |

Когда клиент создает сеанс, роли, назначенные для сеанса, зависят от правил, определенных для каждой роли. В таблице 3 приведен список назначенных ролей для сеансов, созданных пользователями, клиентскими приложениями и конечными точками.

Таблица 3 — Пример назначения роли

| Предоставленный клиентом пользователь | Назначенные для сеанса роли |
|---|---|
| Аноним | Аноним |
| Сэм | Подтвержденный пользователь |
| Джо, использующий приложение станция оператора 1 | Подтвержденный пользователь, оператор 1 |
| Джо, использующий приложение станция оператора 2 | Подтвержденный пользователь, оператор 2 |
| Джо, использующий общее приложение | Подтвержденный пользователь |
| Корень, использующий приложение станция оператора 1 | Подтвержденный пользователь, руководитель |
| Корень, использующий приложение и конечную точку 127.0.0.1 | Подтвержденный пользователь, руководитель, администратор |
| Корень, использующий приложение и другую конечную точку | Подтвержденный пользователь, руководитель |

Когда клиентское приложение получает доступ к узлу, разрешения роли для узла сравниваются с ролями, назначенными сеансу. Любое разрешение, доступное хотя бы на одну роль, предоставляется клиенту.

# 5 Стандартные классы узла

## 5.1 Обзор

Раздел 5 определяет классы узла, используемые для определения узлов в адресном пространстве OPC UA. Классы узла производятся от общего базового класса узла. Сначала определяется класс узла, затем классы узла, используемые для организации адресного пространства, и наконец классы узла, используемые для представления объектов.

Классы узла, определенные для представления объектов, делятся на три категории применения: для определения экземпляров, для определения типов этих экземпляров и для определения типов данных.

## 5.2 Базовый класс узла

### 5.2.1 Общие положения

Модель адресного пространства OPC UA определяет базовый класс узла, из которого выводятся все остальные классы узла. Полученные классы узла представляют различные компоненты модели объекта OPC UA (см. 4.2). Атрибуты базового класса узла приведены в таблице 4. Для базового класса узла не указано ссылок.

Таблица 4 — Базовый класс узла

| Имя | Использование | Тип данных | Описание |
|---|---|---|---|
| Атрибуты | | | |
| Идентификатор узла | M | Идентификатор узла | См. 5.2.2 |
| Класс узла | M | Класс узла | См. 5.2.3 |
| Поисковое имя | M | Квалифицированное имя | См. 5.2.3 |
| Отображаемое имя | M | Локализованный текст | См. 5.2.4 |
| Описание | O | Локализованный текст | См. 5.2.5 |
| Маска записи | O | Атрибут маски записи | См. 5.2.6 |

*Окончание таблицы 4*

| Имя | Использование | Тип данных | Описание |
|---|---|---|---|
| Маска записи пользователя | O | Атрибут маски записи | См. 5.2.7 |
| Разрешения роли | O | Тип разрешения роли | См. 5.2.8 |
| Разрешения роли пользователя | O | Тип разрешения роли | См. 5.2.9 |
| Ограничения доступа | O | Тип ограничений доступа | См. 5.2.10 |
| Ссылки | | | Для этого класса узла ссылок не указано |

**5.2.2 Идентификатор узла**

Узлы однозначно идентифицируются с помощью построенного идентификатора, называемого идентификатором узла. Сервер должен сохранять идентификатор, принадлежащий узлу, то есть не должен генерировать новые идентификаторы узла во время перезагрузки.

**5.2.3 Имя узла**

Атрибут класса узла идентифицирует класс узла самого узла. Узлы имеют атрибут поискового имени, который используется как нелокализованное имя, адаптированное для чтения человеком при просмотре адресного пространства для создания путей из поисковых имен. Служба перевода путей поиска в идентификаторы узлов, определенная в ГОСТ Р 71809, может быть использована для следования путем, созданным из поисковых имен.

Поисковое имя не должно использоваться для отображения имени узла. Вместо этого следует использовать отображаемое имя.

В отличие от идентификаторов узла поисковое имя нельзя использовать для однозначной идентификации узла. Разные узлы могут иметь одно и то же поисковое имя. Структура поискового имени содержит пространство имен и строку. Пространство имен необходимо для того, чтобы сделать поисковое имя уникальным в контексте узла (например, свойства узла), хотя и не уникальным в контексте сервера. Если различные организации определяют поисковые имена для свойств, пространство поисковых имен, предоставляемое организацией, делает поисковое имя уникальным, хотя разные организации могут использовать одну и ту же строку, имеющую слегка разное значение.

Серверы часто могут использовать одно и то же пространство имен для идентификатора узла и поискового имени. Однако, если они хотят предоставить стандартное свойство, его поисковое имя должно обладать пространством имен органа по стандартизации, пусть пространство имен идентификатора узла и отражает что-то другое, например локальный сервер.

Часть поискового имени со строкой чувствительна к регистру, то есть клиенты должны считать их чувствительными к регистру. Серверы обрабатывают поисковые имена, передаваемые в запросах служб, как не зависящие от регистра. Например, служба перевода путей поиска в идентификаторы узлов или фильтр событий.

**5.2.4 Отображаемое имя**

Атрибут отображаемого имени содержит локализованное имя узла. Клиенты должны использовать этот атрибут, если они хотят отобразить узел пользователю. Для этой цели они не должны использовать поисковое имя. Сервер может поддерживать одно или несколько локализованных представлений для каждого отображаемого имени. Когда клиенты открывают сеанс с сервером они должны договариваться о возвращении региональных настроек. Для описания установления сеанса и региональных настроек следует руководствоваться ГОСТ Р 71809.

**5.2.5 Описание**

Необязательный атрибут описания должен объяснять значение узла в локализованном тексте с использованием тех же механизмов локализации, которые описаны для отображаемого имени в 5.2.4.

**5.2.6 Маска записи**

Необязательный атрибут маски записи открывает возможности клиента для записи атрибутов узла. Атрибут маски записи не учитывает никаких прав доступа пользователя, то есть хоть атрибут и может быть записан, он может быть ограничен определенным пользователем/группой пользователей (см. 5.2.10)

Если сервер OPC UA не имеет возможности получить информацию о маске записи для определенного атрибута из базовой системы, он должен заявить, что он доступен для записи [3]. Если операция записи вызывается в атрибуте, сервер должен передать этот запрос и вернуть соответствующий код состояний, если такой запрос будет отклонен. Коды состояний определены в ГОСТ Р 71809.

**5.2.7 Маски записи пользователя**

Необязательный атрибут маски записи пользователя раскрывает возможности клиента по записи атрибутов узла с учетом прав доступа пользователя. Клиент использует тип данных атрибута записи пользователя, определенный в [1].

Атрибут маски записи пользователя далее может только ограничивать атрибут маски записи, когда он установлен как не записываемый в общем случае, который применяется для каждого пользователя. Клиенты могут не предполагать, что атрибут может быть написан на основе атрибута маски записи пользователя. Возможно, что сервер может вернуть ошибку об отказе в доступе из-за какого-то особого для сервера изменения, которое не было отражено в состоянии данного атрибута на тот момент, когда клиент к нему обращался.

**5.2.8 Разрешения роли**

Необязательный атрибут разрешений роли определяет разрешения, которые применяются к узлу для всех ролей, имеющих к нему доступ. Значение атрибута является массивом структур типов разрешений роли (см. таблицу 5).

Таблица 5 — Тип разрешения роли

| Имя | Тип | Описание |
|---|---|---|
| Тип разрешения роли | Структура | Определяет разрешения для роли |
| Идентификатор роли | Идентификатор роли | Идентификатор узла объекта роли |
| Разрешения | Тип разрешения | Маска, указывающая, какие разрешения доступны для роли |

Серверы разрешают администраторам писать в атрибуте разрешений роли [2].

Если это не указано, вместо него должно использоваться значение свойства разрешений роли по умолчанию из объекта пространства имен метаданных, связанного с узлом. Если объект пространства имен метаданных не определяет свойство или не существует, то сервер не должен публиковать никакой информации о том, как он управляет разрешениями.

Если сервер поддерживает разрешения для определенного пространства имен, он должен по умолчанию добавить свойство разрешений роли в объект пространства имен метаданных для этого пространства имен (см. рисунок 6). Если определенному узлу в пространстве имен необходимо переопределить значения по умолчанию, сервер добавляет атрибут разрешения роли к узлу. Свойство разрешений роли по умолчанию и атрибуты разрешения роли должны быть доступны для чтения только администраторам. Если сервер позволяет изменять разрешения, то эти значения должны быть записаны. Если сервер позволяет переопределять разрешения для определенного узла, но на данный момент не имеет настроенных разрешений узла, то значение атрибута должно быть пустым массивом. Если администратор желает удалить переопределенные разрешения, то в этот атрибут записывается пустой массив. Серверы должны предотвращать изменение разрешений таким образом, чтобы сделать сервер неработоспособным.

Если сервер публикует информацию о ролях для пространства имен, назначенных для текущего сеанса, он должен добавить свойство разрешения роли пользователя по умолчанию в объект пространства имен метаданных для этого пространства имен. Значение этого свойства должно представлять собой читаемый список разрешений на каждую роль, назначенную на текущий сеанс. Если конкретный узел в пространстве имен переопределяет разрешения роли по умолчанию, сервер также переопределяет разрешения роли пользователя по умолчанию путем добавления атрибута разрешений роли пользователя на узел. Если сервер позволяет переопределить разрешения для определенного узла, но на данный момент не имеет настроенных разрешений узла, то сервер должен вернуть значение свойства разрешений роли пользователя по умолчанию для пространства имен узла.

Если сервер реализует специфическую для изготовителя модель разрешения роли для пространства имен, он не должен добавлять разрешения роли по умолчанию или свойства разрешений роли пользователя по умолчанию к пространству имени объекта метаданных.

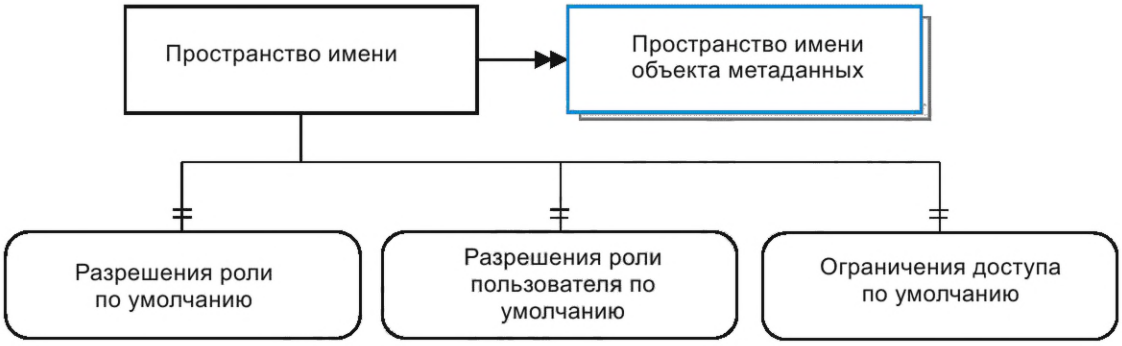


Рисунок 6 — Разрешения в адресном пространстве

**5.2.9 Разрешения роли пользователя**

Необязательный атрибут разрешений роли пользователя указывает разрешения, которые применяются к узлу для всех ролей, предоставленных для текущего сеанса. Значение атрибута представляет собой массив структур типа разрешений роли (см. таблицу 5).

Клиенту допускается определять свои эффективные разрешения, выполнив логическую операцию «ИЛИ» из разрешений для каждой роли в массиве. Значение этого атрибута складывается из правил, используемых сервером для сопоставления сеансов с ролями. Данное отображение является специфическим для конкретного поставщика или использует стандартную модель роли.

Данный атрибут не должен быть доступен для записи.

Если не указано иное, вместо этого используется значение свойства разрешений роли пользователя по умолчанию из пространства имени объекта метаданных, связанного с узлом. Если пространство имен объекта метаданных не определяет свойство или не существует, то сервер не публикует никакой информации о ролях, привязанных к текущему сеансу [3].

**5.2.10 Ограничения доступа**

Необязательный атрибут ограничений доступа определяет ограничения доступа, применимые к узлу. Их тип данных определен в [4]. Если сервер поддерживает ограничения доступа для определенного пространства имен, он добавляет свойство ограничения доступа по умолчанию в пространство имен объекта метаданных для этого пространства имен (см. рисунок 6). Если определенному узлу в пространстве имен необходимо переопределить значение по умолчанию, сервер добавляет атрибут ограничений доступа к узлу.

Если сервер реализует конкретную модель ограничения доступа, заданную поставщиком для пространства имен, он не добавляет свойство ограничения доступа по умолчанию к пространству имен объекта метаданных [5].

### 5.3 Класс узла типа ссылки

#### 5.3.1 Общие положения

Ссылки определены как экземпляры узлов типа ссылки. Узлы типа ссылки видны в адресном пространстве и определяются с помощью класса узла типа ссылки, как указано в таблице 6. Напротив ссылка является неотъемлемой частью узла, а для представления ссылок не используется класс узла.

Настоящий стандарт определяет набор типов ссылок, предоставляемых как неотъемлемая часть модели адресного пространства OPC UA. Эти типы ссылок определены в 5.3.3, а их представление в адресном пространстве определено в ГОСТ Р 71810. Серверы могут также определять типы ссылок. ГОСТ Р 71809 определяет службы управления узлами, которые позволяют клиентам добавлять типы ссылок в адресное пространство.

Таблица 6 — Класс узла типа ссылки

| Имя | Использование | Тип данных | Описание |
|---|---|---|---|
| Атрибуты | | | |
| Базовые атрибуты класса узла | M | — | Унаследованная от базового класса узла, см. 5.2 |

*Окончание таблицы 6*

| Имя | Использование | Тип данных | Описание |
|---|---|---|---|
| Является абстрактным | M | Булев тип | Булев атрибут со значениями:<br>True: абстрактный тип ссылки, т. е. не должно существовать ссылки этого типа, лишь его подтипы;<br>False: не абстрактный тип ссылки, т. е. ссылки этого типа могут существовать |
| Симметричность | M | Булев тип | Булев атрибут со значениями:<br>True: значение типа ссылки такое же, как и у узла источника и целевого узла;<br>False: значение типа ссылки, как видно из целевого узла, обратно пропорционально, как видно из узла источника |
| Обратное имя | O | Локализованный текст | Обратное имя ссылки, которое является значением типа ссылки, как видно из целевого узла |
| Ссылки | | | |
| Наличие свойства | 0…* | — | Используется для идентификации свойств (см. 5.3.3.2) |
| Наличие подтипа | 0...* | — | Используется для идентификации свойств (см. 5.3.3.3) |
| Стандартные свойства | | | |
| Версия узла | O | String | Свойство версии узла используется для указания версии узла.<br>Свойство версии узла обновляется, когда ссылка добавляется в узел или удаляется из узла, которому принадлежит свойство. Изменение значений атрибутов не приводит к изменению версии узла. Клиенты могут прочитать свойство версии узла или подписаться на него, чтобы определить, когда структура узла была изменена |

### 5.3.2 Атрибуты

Класс узла типа ссылки наследует базовые атрибуты базового класса узла, определенного в 5.2. Наследуемый атрибут поискового имени используется для указания типа ссылки, как видно из узла источника. Например, тип ссылки с поисковым именем «содержание» используется в ссылках, в которых указано, что источник узла содержит целевой узел. Наследуемый атрибут отображаемого имени содержит перевод поискового имени.

Поисковое имя типа ссылки должно быть уникальным на сервере. Не допускается, чтобы два разных типа ссылок имели одно и то же поисковое имя ([2], ГОСТ Р 71810).

Атрибут «является абстрактным» указывает, является ли тип ссылки абстрактным. Абстрактные типы ссылки не могут быть конкретизированы и используются только по организационным причинам, например для указания некоторой общей семантики или ограничений, которые наследуют подтипы.

Атрибут симметричности используется для указания того, одинаково ли значение типа ссылки для узла источника и целевого узла.

Если тип ссылки симметричен, то атрибут обратного имени опускается. Примерами симметричных типов ссылок являются «подключение к» и «связь с». Оба подразумевают одну и ту же семантику, исходящую от узла источника или целевого узла. Поэтому оба направления рассматриваются как прямые ссылки.

Если тип ссылки является несимметричным и не абстрактным, то должен быть установлен атрибут обратного имени. Атрибут обратного имени определяет значение типа ссылки, как видно из целево-

го узла. Примеры несимметричных типов ссылки включают в себя «содержит» и «содержится», а также «получает от» и «отправляет в».

Ссылки, использующие обратное имя, например ссылки «содержится», называются обратными ссылками.

Для серверов не всегда возможно реализовать прямые и обратные ссылки для несимметричных типов ссылок, как показано на рисунке. При этом ссылки называются двунаправленными. Хотя это и не требуется, рекомендуется, чтобы все иерархические ссылки были конкретизированы как двунаправленные для обеспечения связи при просмотре. Двунаправленная ссылка моделируется как две отдельные ссылки. На рисунке 7 приведены примеры симметричных и несимметричных ссылок, а также использования обзора имен и обратного имени, см. ГОСТ Р 70988.

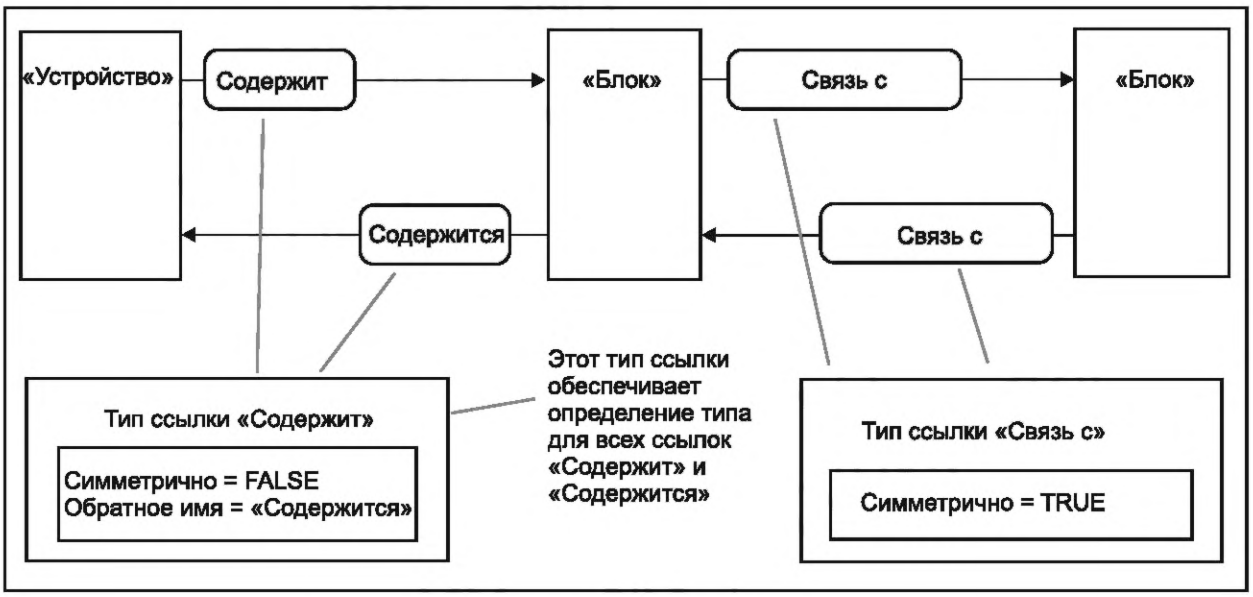


Рисунок 7 — Симметричные и несимметричные ссылки

В качестве примера однонаправленной ссылки часто бывает так, что приемник сигнала знает свой источник сигнала, но этот источник сигнала не знает своего приемника сигнала. Приемник сигнала будет иметь ссылку «из источника» на источник сигнала, при этом источник сигнала не будет иметь соответствующие обратные ссылки «из источника в».

Отображаемое имя и обратное имя являются единственными стандартизированными местами, указывающими семантику типа ссылки. Может существовать более сложная семантика, связанная с типом ссылки, чем может быть выражена в этих атрибутах (например, семантика наличия подтипа). В настоящем стандарте не уточняется, как должна быть раскрыта эта семантика. Однако для этой цели можно использовать атрибут описания. Данный стандарт содержит семантику типов ссылок, указанных в 5.3.3.

Тип ссылки может иметь ограничения его использования. Например, можно указать, что, начиная с узла А и только следуя ссылкам на этот тип ссылки или один из его подтипов, он не сможет вернуться к А. Это ограничение «без цикла».

В данном стандарте не указано, как эти ограничения могут и должны быть доступны в адресном пространстве. Данный стандарт не ограничивает тип ограничений, действительных для типа ссылок. Это может, например, также повлиять на тип объекта. Ограничение, заключающееся в том, что тип ссылки может использоваться только путем связывания узлов некоторых классов узлов с определенной мощностью отношений, является особым ограничением типа ссылки.

**5.3.3 Ссылки**

5.3.3.1 Общие положения

Ссылки наличия подтипа и ссылки наличия свойства являются единственными типами ссылок, которые могут использоваться с типом ссылки узлов в качестве узла источника. Узлы типа ссылок не должны быть узлами источника других типов ссылок.

5.3.3.2 Ссылки наличия свойства

Ссылки наличия свойства используются для идентификации свойств типа ссылки и должны относиться только к узлам переменной класса узла.

Свойство версии узла используется для указания версии типа узла.

В настоящем стандарте не определены дополнительные свойства для типов ссылки. Дополнительные части системы стандартов OPC UA могут определять дополнительные свойства для типов ссылок.

5.3.3.3 Ссылки наличия подтипа

Ссылки наличия подтипа используются для определения подтипов типов ссылок. Не требуется предоставлять ссылку наличия подтипа для супертипа, но подтип должен предоставить обратную ссылку на супертип. Для подтипа применяются следующие правила:

а) Семантика типа ссылок (например, «охватывает иерархию») наследуется в его подтипах и может быть уточнена там (например, «охватывает специальную иерархию»). Отображение имени, а также обратное имя для несимметричных типов ссылок отражают специализацию.

б) Если тип ссылки указывает некоторые ограничения (например, «не допускать циклов»), то он наследуется и может быть только уточнен (например, наследование «нет циклов» может быть уточнено как «должно быть деревом — только один родитель»), но не понижен (например, «разрешать циклы»).

в) Ограничения, на которые можно ссылаться классам узла, также наследованы и могут быть дополнительно ограничены. Если тип ссылки «А» не позволяет связать объект с типом объекта, это также верно и для его подтипов.

г) Тип ссылки должен иметь ровно один супертип, за исключением ссылки типа ссылки, определенного в 7.2, как корневой тип иерархии типа ссылки. Иерархия типа ссылки не поддерживает множественное наследование [6].

### 5.4 Представление класса узла

Базовые системы часто являются крупными, и клиенты часто заинтересованы только в конкретном подмножестве данных. Им не нужно или они не хотят просматривать узлы в адресном пространстве, к которым они не имеют интереса.

Все узлы, содержащиеся в представлении, должны быть доступны, начиная с представления узла при просмотре в контексте представления. Не ожидается, что все узлы могут быть просмотрены непосредственно из представления узла и из других узлов, содержащихся в представлении.

Представлению узла допустимо применение не только как дополнительной точки входа в адресное пространство, но и как конструкции для организации адресного пространства и, таким образом, как единственной точки входа в подмножество адресного пространства. Клиенты не должны игнорировать представления узлов при открытии адресного пространства. Простые клиенты, которые не работают с представлениями для целей фильтрации, могут работать с представлением узла как с объектом типа папки (см. 5.5.3).

Представление класса узла наследует базовые атрибуты от базового класса узла, определенного в 5.2. Он также определяет два дополнительных атрибута.

Обязательный атрибут «не содержит циклов» имеет значение «False», если сервер не может определить, содержит ли представление петли или нет.

Обязательный атрибут уведомителя события идентифицирует, возможно ли использовать представление для подписки на события, которые происходят либо в содержимом данного представления, либо как изменение модели события, либо для чтения/записи истории событий.

Представление, которое поддерживает события, должно предоставлять все события, происходящие в любом объекте, используемом как уведомитель события, которое является частью содержимого представления. Кроме того, он должен предоставлять все изменения модели события, которые происходят в контексте представления.

Чтобы избежать рекурсии, т. е. получения всех событий сервера, объект сервера, определенный в ГОСТ Р 71810, не должен быть частью какого-либо представления, так как он предоставляет все события сервера.

Данный стандарт определяет понятие представления. Каждое представление определяет подмножество узлов в адресном пространстве. Все адресное пространство является представлением по умолчанию. Каждый узел в представлении может содержать только подмножество его ссылок, опреде-

ленных создателем представления. Представления определяются в виде класса узла, что указано в таблице 7.

Таблица 7 — Представление класса узла

<table>
<tr><th>Имя</th><th>Использование</th><th>Тип данных</th><th>Описание</th></tr>
<tr><td colspan="4">Атрибуты</td></tr>
<tr><td>Атрибуты базового класса узла</td><td>M</td><td>—</td><td>Наследованная от базового узла класса, см. 5.2</td></tr>
<tr><td>Не содержит циклы</td><td>M</td><td>Булево значение</td><td>Если установить значение «True», то этот атрибут указывает на то, что, следуя ссылкам в представлении, петель не будет, т. е. начиная с узла «A», содержащегося в представлении, и следующий перед ссылками в контексте узла представления «A», не будет достигаться. Он не указывает, что есть только один путь, начинающийся с узла представления, чтобы достичь узла, содержащегося в представлении.<br>Если установить значение «False», то этот атрибут указывает на то, что следующие ссылки в контексте представления могут приводить к возникновению циклов</td></tr>
<tr><td>Уведомление о событии</td><td>M</td><td>Byte</td><td>Атрибут уведомителя события используется для указания, можно ли использовать узел для подписки на события или для чтения/записи исторических событий.<br>Уведомитель события — это 8-битное беззнаковое целое число со структурой, определенной в следующей таблице
<table>
<tr><th>Поле</th><th>Бит</th><th>Описание</th></tr>
<tr><td>Подписка на события</td><td>0</td><td>Указывает, можно ли использовать его для подписки на события (0 означает, что не может быть использовано для подписки на события, 1 означает, что может быть использовано для подписки на события)</td></tr>
<tr><td>Резервный</td><td>1</td><td>Зарезервировано для будущего использования. Всегда будет ноль</td></tr>
<tr><td>Чтение истории</td><td>2</td><td>Показывает, читается ли история событий.<br>(0 — нечитаемый, 1 — читаемый)</td></tr>
<tr><td>Запись истории</td><td>3</td><td>Указывает, можно ли историю событий записывать (0 — не записываемый, 1 — записываемый)</td></tr>
<tr><td>Резервный</td><td>4:7</td><td>Зарезервировано для будущего применения. Всегда должен быть равен нулю</td></tr>
<tr><td colspan="3">Вторые два бита указывают доступность истории событий через сервер OPC UA</td></tr>
</table></td></tr>
<tr><td colspan="4">Ссылки</td></tr>
<tr><td>Иерархические ссылки</td><td>0...*</td><td></td><td>Узлы верхнего уровня в представлении ссылаются на иерархические ссылки (см. [7])</td></tr>
</table>

*Окончание таблицы 7*

| Имя | Использование | Тип данных | Описание |
|---|---|---|---|
| Наличие свойства | 0…* | | Ссылки на свойства идентифицируют свойства представления |
| Стандартные свойства | | | |
| Версия узла | O | | Свойство версии узла используется для указания версии узла.<br>Свойство версии узла обновляется каждый раз, когда ссылка добавляется в узел или удаляется из узла, которому принадлежит это свойство. Изменение значений атрибутов не приводит к изменению версии узла. Клиенты могут прочитать свойство версии узла или подписаться на него, чтобы определить, когда структура узла изменилась |
| Версия представления | O | Unit32 | Номер версии для представления. При добавлении или удалении узлов из представления значение свойства версии представления обновляется. Клиенты могут обнаружить изменения в составе представления, используя это свойство. Значение версии представления всегда должно быть больше нуля |

Представления определяются сервером. Службы просмотра и запроса, определенные в ГОСТ Р 71809, ожидают, что идентификатор узла в представлении узла обеспечит эти службы в контексте представления.

Ссылки наличия свойства используются для идентификации свойств представления. Свойство версии узла используется для указания версии узла. Свойство версии представления указывает версию содержимого представления. В отличие от версии узла свойство версии представления обновляется, даже если узлы, на которые нет прямых ссылок в представлении узла, добавляются или удаляются из представления. Это свойство является необязательным, поскольку серверы могут быть не в состоянии обнаружить изменения в содержимом представления. Серверы могут также генерировать изменение модели события, если узлы добавляются или удаляются из представления. В данном стандарте не определены дополнительные свойства для представлений. Дополнительные части серии стандартов OPC UA могут определять дополнительные свойства для представлений.

Представления могут быть узлом источника любой иерархической ссылки. Они не должны быть узлом источника какой-либо неиерархической ссылки.

### 5.5 Объекты

#### 5.5.1 Класс узла объекта

Объекты используются для представления систем, компонентов системы, реальных объектов и программных объектов. Объекты определяются с помощью класса узла объекта,

Объект класса узла наследует базовые атрибуты от базового класса узла, определенного в 5.2.

Обязательный атрибут уведомитель событий определяет, уточняя можно ли использовать объект для подписки на события или для чтения и записи истории событий.

Объект класса узла использует ссылку на наличие компонента для определения переменных данных, объектов и методов объекта.

Он использует ссылку на наличие свойства для определения свойства объекта. Свойство версии узла используется для указания версии объекта. Свойства пиктограммы предоставляют пиктограмму объекта. Свойство правила наименования определяет правило наименования у правила моделирования и должно применяться только к объектам типа правила моделирования. Дополнительные свойства для объектов определены в ГОСТ Р 71811. Дополнительные части серии стандартов OPC UA могут определять дополнительные свойства объектов.

Для указания своего правила моделирования объект может использовать максимум одну ссылку на наличие правила моделирования, указывающую на объект правила моделирования. Правила представления узла определены в 5.4.

Ссылки на наличие уведомления и наличие источника события используются для предоставления информации о событиях и могут быть применены только к объектам, используемым в качестве уведомителей событий, см. ГОСТ Р 59799.

Ссылка на определение наличия типа указывает на тип объекта, используемый в качестве определения типа объекта.

Объекты могут использовать любые дополнительные ссылки для определения связей с другими узлами. На типы используемых ссылок или на классы узлов, на которые могут быть сделаны ссылки, ограничения не накладываются. Однако ограничения могут быть определены типом ссылки, исключающим его использование для объектов.

Если объект используется как декларация экземпляра (см. 4.4.2), то все узлы, на которые сделаны ссылки с прямым направлением иерархических ссылок, должны иметь уникальные поисковые имена в контексте данного объекта, см. [1].

Если объект создан на основе декларации экземпляра, то он должен иметь то же поисковое имя, что и декларация экземпляра.

**5.5.2 Класс узла типа объекта**

Типы объектов дают определения объектам. Типы объектов определяются с помощью класса узла типа объекта.

Класс узла типа объекта наследует базовые атрибуты от базового класса узла, определенного в 5.2. Дополнительный атрибут является абстрактным, указывает, является ли тип объекта абстрактным или нет.

Класс узла типа объекта использует ссылки на наличие компонента для определения переменных данных, объектов и методов.

Ссылка на наличие свойства используется для идентификации свойств. Свойство версии узла используется для указания версии типа объекта. Пиктограмма свойства предоставляет пиктограмму типа объекта. В настоящем стандарте не определены дополнительные свойства для типов объектов. Дополнительные части серии стандартов OPC UA могут определять дополнительные свойства для типов объектов.

Ссылки на наличие подтипа используются для создания подтипов типов объекта. Подтипы типа объекта наследуют общую семантику от родительского типа. Применяются общие правила подтипирования, как определено в разделе 7. Необязательно предоставлять ссылку на наличие подтипа для супертипа, но требуется, чтобы подтип предоставлял обратную ссылку на свой супертип.

Ссылки на генерацию события определяют тип событий, генерируемых экземплярами типа объекта. Эти объекты являются источниками события указанного типа или одного из его подтипов. Серверы должны делать ссылки на генерацию события двунаправленными ссылками. Допускается односторонний переход, когда сервер не может указать обратное направление, т. е. от типа события к каждому типу объекта, поддерживающему тип события. Атрибут уведомителя о событии объекта и ссылки на генерацию события его типа объекта не связаны между собой. Объекты, которые могут генерировать события, могут не использоваться в качестве объектов, на которые клиенты подписываются для получения соответствующих уведомлений о событиях.

Ссылки на генерацию события являются необязательными, т. е. объекты могут генерировать события типа события, которые не открываются его типом объекта.

Тип объекта может использовать любые дополнительные ссылки для определения связей с другими узлами. На типы используемых ссылок или на классы узлов ограничения не накладываются. Ограничения могут быть определены типом ссылки, исключая его использование для типов объектов. Все узлы, на которые имеются прямые иерархические ссылки, должны иметь уникальные поисковые имена в контексте типа объекта (см. 5.5).

**5.5.3 Стандартный тип папки типа объекта**

Тип папки типа объекта формально определен в ГОСТ Р 71810. Целью генератора объекта является предоставление объектов, которые не имеют иной семантики, кроме организации адресного пространства. Для этих объектов-папок вводится специальный тип ссылки — тип ссылки организации. Узлом источника такой ссылки всегда должно быть представление или объект типа папки типа объекта; целевой узел может быть любого класса узла. Ссылка организации может быть использована в любой комбинации со ссылками на наличие дочернего элемента (наличие компонента, наличие свойства и т. д.; см. ГОСТ Р 71806) и не препятствует возникновению петель. Так, они могут быть использованы, чтобы охватить несколько иерархий.

### 5.6 Переменные

#### 5.6.1 Общие положения

Определены два типа переменных: свойства и переменные данных. Несмотря на то, что они отличаются способом использования, описанным в 4.4, и имеют различные ограничения, описанные в 5.2.10. Переменные используются для представления значений, могущих быть простыми или сложными. Переменные определяются типами переменных и представляются согласно разрешениям роли пользователя (см. 5.2.9).

Переменные всегда определяются как свойства или переменные данных других узлов в адресном пространстве. Переменная является частью хотя бы одного узла, но может быть связана с любым количеством других узлов. Переменные определяются с помощью переменных класса узла. Переменная класса узла наследует базовые атрибуты от базового класса узла, определенного в 5.2. Атрибут значения представляет собой значение переменной. Атрибуты типа данных, рейтинга значения и размерностей массива предоставляют возможность описания простых и сложных значений.

#### 5.6.2 Свойства

Свойства используются для определения характеристик узлов. Свойства определяются с помощью переменной класса узла. Однако, свойства ограничивают использование переменных.

Свойства — лист любой иерархии, они не должны быть узлом источника любых иерархических ссылок. Сюда входит ссылка на наличие компонента или ссылка на наличие свойства, то есть свойства не содержат свойств и не могут раскрыть их сложную структуру. Однако они могут быть узлом источника любых неиерархических ссылок [2].

Ссылка на наличие определения типа указывает на тип переменной свойства. Поскольку свойства уникально идентифицируются по поисковому имени, все свойства должны указывать на тип свойства, определенный в ГОСТ Р 71810.

Свойства всегда должны определяться в контексте другого узла и быть узлом источника хотя бы одной ссылки на наличие свойства. Чтобы отличить их от переменных данных, они не должны быть целевым узлом любой ссылки на наличие компонента. Поисковое имя свойства всегда уникально в контексте узла. Запрещается, чтобы узел ссылался на две переменные, используя ссылки на наличие свойства, имеющие одно и то же поисковое имя.

#### 5.6.3 Класс узла типов переменной

Типы переменных используются для определения типов переменных. Типы переменных определяются с помощью класса узла типов переменной. Класс узла типов переменной наследует базовые атрибуты класса узла, определенные в 5.2. Класс узла типа переменной также определяет набор атрибутов, описывающих стандартное или начальное значение его экземпляра переменных. Атрибут значения представляет собой значение по умолчанию. Атрибуты типа данных, рейтинга значения и размерностей массива предоставляют возможность описания простых и сложных значений. Атрибут определяет, может ли тип быть конкретизирован.

Класс узла типов переменной использует ссылки наличия свойства для определения свойств и ссылок наличия компонентов для определения переменных данных. От правил моделирования. зависит будут ли они конкретизированы.

Свойство версии узла указывает версию типа переменной. В настоящем стандарте не определены дополнительные свойства для типов переменных. Дополнительные части системы стандартов OPC UA определяют дополнительные свойства для типов переменных. [1] определяет набор свойств, которые могут использоваться для типов переменных.

Ссылки наличия подтипа используются для подтипа типов переменных. Подтипы типов переменных наследуют общую семантику от родительского типа. Общие правила подтипирования определены в разделе 7. Необязательно предоставлять ссылку наличия подтипа для супертипа, но требуется, чтобы подтип предоставлял обратную ссылку на свой супертип [8].

Ссылки генерации события определяют, что тип переменных может быть источником события, указанного типом события или одного из его подтипов. Серверы должны создавать двунаправленные ссылки генерации события. Ссылкам разрешается быть однонаправленными, когда сервер не может раскрыть обратное направление, указывающее на тип переменной, поддерживающий тип события.

### 5.7 Метод класса узла

Методы определяют вызываемые функции. Методы вызываются с помощью службы вызовов, определенной в ГОСТ Р 71809. Вызовы методов в адресном пространстве не отображаются. Вызовы методов всегда выполняются до завершения и всегда возвращают ответы после завершения.

Метод класса узла наследует базовые атрибуты от базового класса узла, определенного в 5.2. Метод класса узла не определяет дополнительных атрибутов.

Атрибут исполняемого модуля указывает, является ли метод исполняемым, не принимая во внимание права доступа пользователей. Если сервер OPC UA не может получить информацию исполняемого модуля из базовой системы, он должен заявить, что она исполняемая. Если вызывается метод, то сервер должен передать этот запрос и вернуть соответствующий код состояния, даже если такой запрос отклонен. Коды состояний определены в ГОСТ Р 71809.

Атрибут исполняемого модуля пользователя указывает, является ли метод исполняемым, принимая во внимание права доступа пользователя. Если сервер OPC UA не может получить информацию, связанную с правами пользователя, из базовой системы, то он должен использовать то же значение, что и атрибут исполняемого модуля. Атрибут исполняемого модуля пользователя может быть установлен на значении False, даже если атрибут исполняемого модуля установлен на значении True, но он должен быть установлен на значение False, если атрибут исполняемого модуля установлен на значение False. Клиенты не могут предполагать, что метод может быть выполнен на основе атрибута исполняемого модуля пользователя. Возможно, что сервер может вернуть ошибку об отказе в доступе из-за какого-то специфического для сервера изменения, которое не было отражено в состоянии данного атрибута в то время, когда клиент к нему обращался.

Свойства могут быть определены для методов, использующих ссылки наличия свойства. Свойства входных и выходных аргументов задают входные и выходные аргументы метода. Оба содержат массив аргумента типа данных. Пустой массив или свойство, которое не предоставляется, указывает на отсутствие входных или выходных аргументов для метода.

Свойство версии узла указывает версию метода. Для указания своего правила моделирования метод может использовать не более одной ссылки, указывающей на объект правила моделирования.

Ссылки генерации события определяют, что методы могут генерировать событие указанного типа события или одного из его подтипов для каждого вызова метода. При успешном вызове метода сервер имеет возможность генерировать по одному событию для каждого ссылающегося типа события.

Для определения связей с другими узлами методам допускается использование дополнительных ссылок. На типы используемых ссылок или на классы узлов, на которые могут быть сделаны ссылки, ограничений не накладывается. Если вызывается метод, то идентификатор узла одного из этих узлов должен быть помещен в службу вызова, определенную в ГОСТ Р 71809. Если метод используется как декларация экземпляра, то все узлы, на которые ссылаются с прямыми иерархическими ссылками, должны иметь уникальные поисковые имена в контексте этого метода.

### 5.8 Типы данных

#### 5.8.1 Модель типа данных

Модель типа данных используется для определения простых и структурированных типов данных. Каждая переменная и тип переменной указывают своим атрибутом типа данных на узел класса узла типа данных, как показано на рисунке 8.

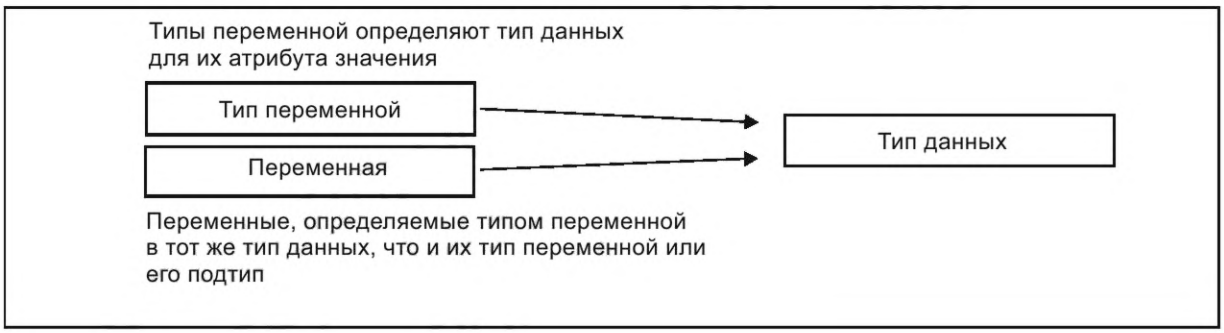


Рисунок 8 — Переменные, типы переменных и их типы данных

Во многих случаях идентификатор узла типа данных — идентификатор типа данных — хорошо известен клиентам и серверам. В разделе 8 определены типы данных, а в ГОСТ Р 71811 — их идентификаторы типа данных. Хорошо известные идентификаторы типа данных обеспечивают общность между серверами OPC UA и позволяют клиентам интерпретировать значения без необходимости считывать с сервера описание типа. В других случаях типы данных и соответствующие им идентификаторы типа данных могут быть определены производителем.

На рисунке 9 показаны узлы, используемые в адресном пространстве для описания структуры типа данных. Тип данных указывает на объект типа кодирования типа данных. Каждый тип данных может иметь несколько кодирований типов данных, например «по умолчанию», «двоичная система исчисления UA» и «XML». Службы, предусмотренные ГОСТ Р 71809, позволяют клиентам запрашивать кодировку или выбирать кодировку «по умолчанию». Каждое кодирование типа данных используется ровно одним типом данных, т. е. не допускается, чтобы два типа данных указывали на одно и то же кодирование типа данных.

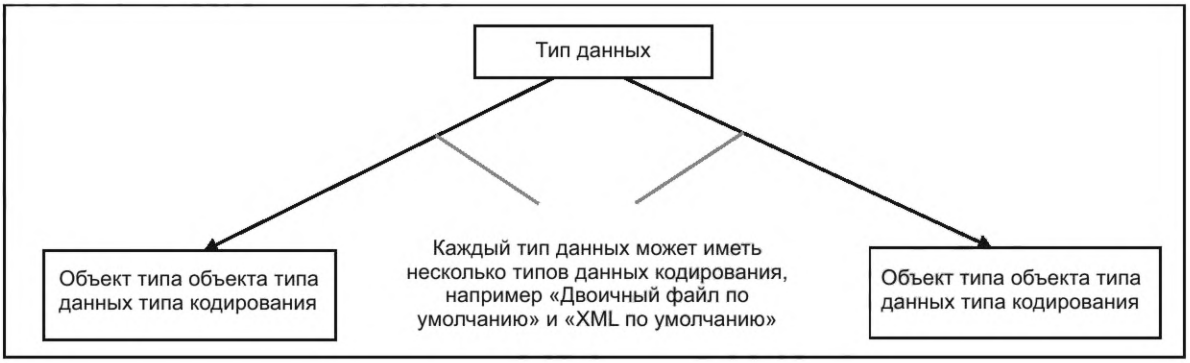


Рисунок 9 — Модель типа данных

Так как идентификатор узла кодирования типа данных будет использоваться в некоторых сопоставлениях для идентификации типа данных и его кодировки, как определено в ГОСТ Р 71811, эти идентификаторы узла могут быть также известны для известных идентификаторов типов данных.

**5.8.2 Правила кодирования для различных видов типов данных**

Встроенные типы данных — фиксированный набор типов данных (полный список встроенных типов дан в ГОСТ Р 71811). Они не имеют кодировок, видимых в адресном пространстве, так как кодировка должна быть известна всем продуктам OPC UA.

Простые типы данных — подтипы встроенных типов данных. Примером простого типа данных является длительность. Она обрабатывается на проводе как число двойственной точности, но клиент может прочитать атрибут типа данных и таким образом интерпретировать значение, определенное длительностью.

Структурированные типы данных — типы данных, которые представляют собой структурированные данные и не определяются, как встроенные типы данных. Структурированные типы данных наследуются прямо или косвенно от структуры типа данных, они могут иметь несколько кодировок, и кодировки выставляются в адресном пространстве. В ГОСТ Р 71811 определено кодирование структурированных типов данных на проводе.

Типы данных перечисления — это типы данных, которые представляют собой дискретные наборы именованных значений. Перечисления всегда кодируются как целое число 32 на проводе, как определено в ГОСТ Р 71811. Типы данных перечисления наследуются прямо или косвенно от типов данных перечисления (см. [1]). Перечисления не имеют кодировок, выставленных в адресном пространстве. Для отображения, читаемого человеком представления перечисленного значения, узел типа данных может иметь свойство строки перечисления, содержащее массив локализованного текста. Целочисленное представление значения перечисления указывает на позицию внутри этого массива. Свойство значений типа перечисления может использоваться вместо строк перечисления для поддержки целочисленного представления перечислений, которые не основаны на нулях или имеют пробелы. Он содержит массив структурированного типа данных, содержащий целочисленное представление, а также

читаемое человеком представление. В дополнение к типам данных, описанным выше, поддерживаются также абстрактные типы данных, которые не имеют кодировок и не могут обмениваться по проводу. Переменные и типы переменных используют абстрактные типы данных для указания того, что их значение может быть любым из подтипов абстрактного типа данных.

**5.8.3 Класс узла типов данных**

В классе узла типов данных описывается синтаксис значения переменной. Класс узла типа данных наследует базовые атрибуты от базового класса узла, определенного в 5.2. Атрибут является абстрактным. Он указывает, является ли тип данных абстрактным. Абстрактные типы данных используются в адресном пространстве, т. е. переменные и типы переменных указывают своим атрибутам типов данных на абстрактный тип данных.

Ссылки наличия кодирования указывают путь от типа данных к кодированию типа данных. Каждый конкретный структурированный тип данных может указывать на множество кодирований типов данных, но каждое кодирование типов данных должно принадлежать одному типу данных, т. е. не допускается, чтобы два узла типа данных указывали на один и тот же объект кодирования типов данных с помощью ссылок наличия кодирования.

Узлы типа данных не должны быть узлом источника других типов ссылок. Тем не менее они могут быть узлом источника других ссылок.

**5.8.4 Кодирование типа данных и кодирующая информация**

Если узел типа данных подвергается воздействию в адресном пространстве, он должен предоставить свои кодирования типа данных с помощью ссылок наличия кодирования. Эти ссылки должны быть двунаправленными.

Информация о том, как кодировать тип данных, приведен в атрибуте определения типа данных узла типа данных. После предоставления клиентам содержание данного атрибута не может быть изменено, так как клиенты могут постоянно кэшировать эту информацию.

## 6 Модель типа для типов объекта и типов переменных

### 6.1 Определения

**6.1.1 Декларация экземпляра**

Декларация экземпляра — объект, переменная или метод, который ссылается на правило моделирования со ссылкой наличия правила моделирования и является целевым узлом иерархической ссылки из узлов определения типа или другой декларации экземпляра. Тип декларации экземпляра может быть абстрактным, однако, он должен быть конкретного типа.

**6.1.2 Экземпляр без правил моделирования**

Если правила моделирования не существует, то для конкретизирования узел не рассматривается ни как экземпляр типа, ни как подтип.

Если узел, на который ссылаются узлы определения типа, не ссылается на правило моделирования, это указывает на то, что этот узел принадлежит только узлу определения типа, а не экземплярам. Например, узел типа объекта может содержать свойство, описывающее сценарии, в которых данный тип может быть использован. Иерархия декларации экземпляра узла определения типа содержит узел определения типа и все декларации экземпляра, на которые прямо или косвенно ссылаются из узла определения типа, используя прямые иерархические ссылки.

**6.1.3 Поисковой путь**

Все цели прямых иерархических ссылок из узла определения типа должны иметь поисковой путь, который является уникальным в пределах узла определения типа. Любая декларация экземпляра в иерархии декларации экземпляра может быть однозначно идентифицирована последовательностью поисковых имен. Эта последовательность поисковых имен называется поисковым путем.

**6.1.4 Идентификаторы узла деклараций экземпляра**

Все узлы определения типа, указанные в ГОСТ Р 71810, определяют идентификаторы узла для своих деклараций экземпляра и поэтому должны использоваться во всех серверах.

### 6.2 Создание подтипов типов объектов и типов переменной

#### 6.2.1 Обзор

Тип ссылки наличия подтипа определяет подтипы типов. Подтипы могут встречаться только между узлами одного и того же класса узла. Правила подтипирования типов ссылки описаны в 5.3.3.

#### 6.2.2 Атрибуты

Подтипы наследуют значения атрибутов родительского типа, за исключением идентификатора узла наследованные значения атрибута переопределяются подтипом, значения поискового и отображаемого имени должны быть переопределены.

## 7 Стандартные типы ссылок

### 7.1 Общие положения

Данный стандарт определяет типы ссылок как неотъемлемую часть модели адресного пространства OPC UA.

### 7.2 Тип ссылки ссылок

Ссылка на тип ссылки — абстрактный тип ссылки, который применяется только в подтипах.

Нет семантики, связанной с этим типом ссылки. Данный тип является базовым для всех типов ссылок. Все типы ссылок должны быть подтипами этого базового типа ссылки — прямыми или косвенными. Основным назначением данного типа ссылки является разрешение простых фильтров и запросов в соответствующих службах согласно ГОСТ Р 71810.

### 7.3 Тип ссылки иерархических ссылок

Тип ссылки иерархических ссылок — абстрактный тип ссылки, который может использоваться только в подтипах.

Семантика иерархических ссылок состоит в том, чтобы обозначить, что ссылки иерархических ссылок охватывают иерархию. Не допускается, чтобы узел источника и целевой узел ссылки типа ссылки иерархической ссылки были одинаковыми, т. е. не допускается наличие самоназваний с помощью иерархических ссылок.

### 7.4 Тип неиерархической ссылки

Тип неиерархической ссылки — абстрактный тип ссылки, который может быть использован только в подтипах.

Семантика неиерархических ссылок в том, чтобы обозначить, что ее подтипы не распространяются на иерархию и не должны соблюдаться при попытке представить иерархию. Для разграничения иерархических и неиерархических ссылок все типы ссылок должны наследоваться.

### 7.5 Тип ссылки агрегации

Тип ссылки агрегации — абстрактный тип ссылки, который может использоваться только в подтипах. Для подтипа наличия дочернего компонента семантикой является указание на то, что часть (целевой узел) принадлежит узлу источника. В нем не указано право собственности на целевой узел.

### 7.6 Тип ссылки наличия компонента

Тип ссылки наличия компонента — конкретный тип ссылки, который можно применять непосредственно. Для данного подтипа типа ссылки агрегации семантика — часть отношений. Целевой узел типа ссылки наличия компонента является частью узла источника. Данный тип ссылки применяется для связи объектов или типов объектов с содержащими их переменными данных и методами.

Целевым узлом этого типа ссылок должна быть переменная, объект или метод.

### 7.7 Тип ссылки наличия свойства

Тип ссылки наличия свойства — конкретный тип ссылки, который можно применять непосредственно. Для данного подтипа типа ссылки агрегации семантикой является определение свойств узла.

Свойства описаны в 5.6.2. Узлом источника данного типа ссылки может быть любой класс узла. Целевой узел должен быть переменной.

### 7.8 Тип ссылки наличия подтипа

Тип ссылки наличия подтипа — конкретный тип ссылки, который можно применять непосредственно. Для подтипа типа ссылки наличия дочернего компонента семантикой является выражение отношения подтипов. Оно используется для охвата иерархии типа ссылок, семантика которой указана в 5.3.3.

### 7.9 Тип ссылки организации

Тип ссылки организации является конкретным типом ссылки и может быть использован непосредственно. Для данного подтипа иерархических ссылок семантикой типа ссылки является организация узлов в адресном пространстве. Данный тип может быть применен, чтобы охватить несколько иерархий, независимых от любой иерархии, созданной с нецикличными ссылками агрегации.

Узел источника ссылок данного типа должен быть объектом или представлением. Если это объект, то он должен являться объектом типа объекта типа папки или одним из его подтипов (см. 5.5.3).

Целевой узел данного типа ссылки может принадлежать к любому классу узла.

### 7.10 Тип ссылки наличия правила моделирования

Тип ссылки наличия правила моделирования является конкретным типом ссылки и может использоваться непосредственно. Он является подтипом неиерархических ссылок. Семантикой данного типа ссылок является привязка правила моделирования к объекту, переменной или методу. Механизмы правила моделирования даны в [8]. Каждый узел должен быть источником узла не более, чем одной ссылки наличия правила моделирования.

### 7.11 Тип ссылки наличия типа определения

Тип ссылки наличия типа определения является конкретным типом ссылки и может использоваться непосредственно. Связи между типами и экземплярами описаны в 4.5. Каждая переменная и каждый объект должны быть узлом источника ровно одной ссылки наличия типа определения.

### 7.12 Тип ссылки наличия кодирования

Тип ссылки наличия кодирования является конкретным типом ссылки и может применяться непосредственно. Для данного подтипа неиерархических ссылок семантикой типа ссылок является ссылка типа данных кодирования подтипа структуры типа данных.

Целевой узел данного типа ссылок должен быть объектом типа объекта типа данных кодирования или одним из его подтипов (см. [2]).

### 7.13 Генерация события

Тип ссылки генерации события является конкретным типом ссылки и может применяться непосредственно. Это подтип неиерархических ссылок.

Семантика этого типа ссылки состоит в том, чтобы идентифицировать типы событий экземпляров типов объекта или типов переменных, которые могут генерироваться, и методы, которые могут генерироваться при каждом вызове метода.

Целевым узлом данного типа ссылки должен быть тип объекта, представляющий типы события, базовый тип события или один из его подтипов.

### 7.14 Тип ссылки «Всегда генерирует событие»

Тип ссылки «Всегда генерирует событие» является конкретным типом ссылки и может применяться непосредственно. Для данного подтипа генерации события семантикой типа ссылки является определение типов методов событий, генерируемых при каждом вызове метода. Узлом источника ссылок данного типа является метод. Целевым узлом данного типа ссылок должен быть тип объекта, представляющий типы события, базовый тип события или один из его подтипов.

### 7.15 Наличие источника события

Тип ссылки наличия источника события является конкретным типом ссылки и может применяться непосредственно. Для подтипа иерархических ссылок семантикой типа ссылки является соотнесение источников события в иерархической, не петлевой организации. Узлом источника данного типа ссылок должен быть объект, являющийся источником подписки на события. Источником подписки на события является объект, бит которого «Подписка на события» установлен в атрибуте уведомителя события.

Целевым узлом этого типа ссылки является узел любого класса узла, который может генерировать уведомления о событиях по подписке на эталонный источник.

### 7.16 Наличие уведомителя

Тип ссылки наличия уведомителя — конкретный тип ссылки, который может применяться непосредственно. Это подтип наличия источника события.

Семантикой данного типа ссылок является связывание объекта узлов, которые являются уведомлениями, с другим уведомлением объекта узлов. Это — подтип типа ссылки наличия источника события, определенный в 7.15.

## 8 Стандартные типы данных

### 8.1 Общие положения

Раздел 8 определяет типы данных. Их представление в адресном пространстве и в иерархии типа данных указано в ГОСТ Р 71810. В других частях серии стандартов OPC UA допускается указание дополнительных типов данных.

### 8.2 Тип идентификатора

Элемент тип идентификатора идентифицирует тип идентификатора узла, его формат и область действия. Областью действия идентификаторов узла является сервер, в котором он определен. Встроенный тип данных должен содержать квалифицированное имя. Например, возможно применение поискового имени. Данный встроенный тип данных определяет значение, которое является либо True, либо False.

### 8.3 Объединение

Этот абстрактный тип данных является базовым типом данных для всех объединенных типов данных. Тип данных является подтипом структурных типов данных. Все типы данных, наследованные от этого типа данных, имеют специальную обработку для кодирования, как определено в ГОСТ Р 71811. Все типы данных объединения должны наследовать непосредственно от этого типа данных.

### 8.4 Тип ограничения доступа

Подтип типа данных целого числа с определенным свойством набора значений опций используется для определения ограничений доступа узла. Тип ограничения доступа формально определен в таблице 8.

Таблица 8 — Определение типа ограничения доступа

| Имя | Тип | Описание |
|---|---|---|
| SigningRequired | 0 | Клиент может получить доступ к узлу только при использовании защищенного канала, который<br>подписывает все сообщения цифровой подписью. Это не относится к разрешению просмотра, если пользователь не установлен |
| EncryptionRequired | 1 | Клиент может получить доступ к узлу только при использовании защищенного канала, который<br>шифрует все сообщения. Это не относится к разрешению просмотра, если пользователь не установлен |

### 8.5 Тип уровня доступа

Подтип типа данных байта содержит определенное свойство набора значений опций. Он используется для указания того, как можно получить доступ к значению переменной (чтение/записывание) и содержит ли она текущие и/или исторические данные. Тип уровня доступа формально определен в таблице 9.

Таблица 9 — Определение типа уровня доступа

| Имя | Тип | Описание |
|---|---|---|
| CurrentRead | 0 | Указывает, доступно ли текущее значение для чтения и доступно ли текущее значение переменной.<br>(0 означает, что оно недоступно для чтения, 1 означает, что его можно прочитать) |
| CurrentWrite | 1 | Указывает, доступно ли текущее значение для записи. Также указывает, доступно ли текущее значение переменной.<br>(0 означает, что оно недоступно для записи, 1 означает, что оно доступно для записи) |
| HistoryRead | 2 | Указывает, доступна ли для чтения история значений. Также указывает, доступна ли история переменной через сервер OPC UA.<br>(0 означает, что значение не читается, 1 означает, что значение можно прочитать) |
| HistoryWrite | 3 | Указывает, доступна ли история значений для записи, а также указывает, доступна ли история переменной через сервер OPC UA.<br>(0 означает, что недоступна для записи, 1 означает, что доступна для записи) |
| SemanticChange | 4 | Этот флаг устанавливается для свойств, которые определяют семантические аспекты родительского узла свойства и в которых значение свойства и следовательно семантика могут изменяться во время работы.<br>(0 означает, что это не семантика, 1 означает, что это семантика) |
| StatusWrite | 5 | Указывает, доступен ли для записи текущий код состояния значения.<br>(0 означает, что доступен только код состояния Good, 1 означает, что доступен любой код состояния) |
| TimestampWrite | 6 | Указывает, доступна ли для записи текущая исходная временная метка.<br>(0 означает, что доступны только нулевые временные метки, 1 означает, что доступно любое значение временной метки) |
| Reserved | 7 | Зарезервировано для использования в будущем. Всегда должно быть равно нулю |

### 8.6 Рамка записи атрибута

Подтип типа данных целого числа без знака используется для определения ограничений доступа к атрибутам узла. Если бит установлен на 0, это означает, что атрибут не записывается. Если бит установлен на 1, это означает, что атрибут записывается. Если узел не поддерживает определенный атрибут, то соответствующий бит устанавливается на ноль.

## Библиография


[1] МЭК 62541-8 Унифицированная архитектура OPC. Часть 8. Доступ к данным

[2] МЭК 62541-7 Унифицированная архитектура OPC. Часть 7. Профили

[3] МЭК 62541-11 Унифицированная архитектура OPC. Часть 11. Доступ к данным за прошлый период

[4] МЭК 62541-9 Унифицированная архитектура OPC. Часть 9. Сигналы предупреждения и состояния

[5] МЭК 62541-13 Унифицированная архитектура OPC. Часть 13. Агрегаты

[6] МЭК 62541-12 Унифицированная архитектура OPC. Часть 12. Обнаружения и глобальные сервисы

[7] МЭК 62541-14 Унифицированная архитектура OPC. Часть 14. «Издатель — подписчик»

[8] МЭК 62541-10 Унифицированная архитектура OPC. Часть 10. Программы



УДК 621.3.049.75:006.354

ОКС 25.040.40
35.100.05

Ключевые слова: цифровая промышленность, унифицированная архитектура, управление цифровой трансформацией производств, мета-модель унифицированной архитектуры, открытая распределенная система, адресное пространство

Редактор *Н.А. Аргунова*
Технический редактор *И.Е. Черепкова*
Корректор *Р.А. Ментова*
Компьютерная верстка *М.В. Малеевой*

Сдано в набор 20.12.2024. Подписано в печать 09.01.2025. Формат 60×84⅛. Гарнитура Ариал.
Усл. печ. л. 3,72. Уч.-изд. л. 3,16.

Подготовлено на основе электронной версии, предоставленной разработчиком стандарта

Создано в единичном исполнении в ФГБУ «Институт стандартизации»
для комплектования Федерального информационного фонда стандартов,
117418 Москва, Нахимовский пр-т, д. 31, к. 2.


gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru